

Technische Universität München
Fakultät für Informatik, Lehrstuhl 10 (Univ-Prof. Dr. Bode):
Rechnertechnik und Rechnerorganisation

Trust-Based Access Control in Federated Environments

Latifa Boursas

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen
Universität München zur Erlangung des akademischen Grades eines Doktors der
Naturwissenschaften (Dr. rer. nat) genehmigten Dissertation.

Vorsitzende: Univ-Prof. Dr. Hans Michael Gerndt
Prüfer der Dissertation: Univ-Prof. Dr. Heinz-Gerhard Hegering
Univ-Prof. Dr. Arndt Bode

Die Dissertation wurde am 22.12.2008 bei der Technischen Universität
München eingereicht und durch die Fakultät für Informatik
am 20.02.2009 angenommen.

Acknowledgments

A thesis is seldom the result of a single person's isolated efforts. Rather, support and encouragement come from different sources in various ways. My thesis has been influenced first and foremost by the support of many people and the culture in the surroundings where it is written. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

I wrote this thesis while working as a research assistant at the Leibniz Supercomputing Centre, and in the same time as a member of the IntegraTUM Project as well as the Munich Network Management Team.

I owe my most deep gratitude and my warmest thanks to my supervisor, Prof. Dr. Heinz-Gerd Hegering for acting as a father rather than a supervisor. His committed guidance, understanding, encouraging and overly enthusiasm and integral view on research have made a deep impression on me and have provided a very important basis for the present thesis.

My heartfelt thanks are also dedicated to Prof. Dr. Arndt Bode, who gave me the opportunity to take part in the very motivating work and learning environment in the IntegraTUM project, who kept an eye on the progress of my work and cheerfully encouraged me to pursue this work on the basis of earlier versions.

I wish to extend my warmest thanks to Dr. Wolfgang Hommel who provided important suggestions on this work as well as sufficient freedom in the project, especially during the intensive time of writing. His kind support and guidance have been of great value in this study.

The MNM Team as well as the IntegraTUM Team also substantially contributed to the development of this work. Especially the strict and extensive comments and the many discussions had a direct impact on the final form and quality of this thesis.

The chain of my gratitude is definitely incomplete, but I feel a deep sense of gratitude for Silvia Knittl and Patricia Marcu who showed to be kind, mostly helpful and trustful colleagues and friends.

I am grateful to my family and friends for their support during the preparation of this thesis.

Latifa Boursas
München, December 2008

Abstract

Nowadays interorganizational collaborations are evolving into large federated environments interconnecting organizations from all over the world. The relationships among these organizations are basically characterized by the need for competition and cooperation, essentially for sharing resources and services such as computing and storage capabilities. Enhanced autonomy and mobility are one of the key features for a continuous and successful functioning of such environments, allowing, thus, the participating parties to engage in ad-hoc collaborations as the need arises.

The dynamic partnering aspect in such organization networks is, on the one hand, leading to the abolishment of classical spatial and temporal constraints, and consequently, to a greater flexibility in cooperation among organizations. On the other hand, this aspect raises other questions such as how to assess the trustworthiness of unknown potential partners, how to rely on their outcomes and how to make authorization decisions thereupon.

In this thesis, a Trust Based Access Control (TBAC) solution, which aims at addressing fundamental trust issues confronting dynamic federated environments throughout the educational and commercial sectors, is presented.

By means of three basic scenarios, which provide insight into the aspects and different classes of the Circle of Trust (CoT) in federated environment, a set of requirements have been collected, weighted and classified in a form of a criteria catalogue, which in turn serves as a basic reference for the solution design. Additionally, a comprehensive survey of much of the literature that can be found on trust and reputation management in distributed and federated environments has been analyzed with regard to the criteria catalogue.

To compensate the deficiencies and the weaknesses of existing approaches in the management of interorganizational trust relationships, a trust process model as well as a framework for building a CoT among organizations has been investigated to support secure and trustful collaborations between them. Firstly, the trust process model specifies the evolution chain of a trust relationship through different phases, including, Initialization, Management, Validation, Evolution and Auditing.

Secondly, the Framework realizes the different phases of the process model, and consequently, enables the specification of a common set of logical methods and procedures for reasoning about trust from different aspects and dimensions. This investigation primarily distinguishes between at least two classes of trust relationships, Collaboration Trust and Content Quality Trust, which basically develop out of the joint experiences of collaborating with regard to additional aspects and behavior indicators such as Quality of Service (QoS) properties and parameters.

The thesis is concluded by an analysis of a prototype implementation of the TBAC Framework, and a detailed evaluation of the trust computation algorithms in the light of performance criteria such as promptness, accuracy, choice of the trust metric scales as well as several other performance parameters.

Zusammenfassung

Durch Kooperation verschiedener Organisationen entstehen Föderationen, die diese Organisationen nicht selten weltweit miteinander verbinden. In diesen Verbänden sollen die jeweiligen Organisationen wettbewerbsfähig bleiben und dennoch miteinander kooperieren, damit sie Dienste oder Ressourcen, wie etwa Rechen- oder Speicherkapazitäten, gemeinsam nutzen können. Der Schlüssel zum Erfolg dieser Föderationen liegt in der individuellen Autonomie und Anpassungsfähigkeit der beteiligten Organisationen, welche je nach Bedarf partizipieren können.

Die daraus resultierende Dynamik führt zur Aufhebung der bisher räumlich und zeitlich begrenzten Strukturen der Verbundorganisation, also Föderation, und erhöht die Flexibilität im Zusammenspiel der Organisationen. Allerdings ergeben sich auch neue Fragestellungen zur Einschätzung der Vertrauenswürdigkeit unbekannter, potentieller Partner im Hinblick darauf, ob deren eigenen Angaben vertraut wird und wie sich entsprechende Kriterien zur Autorisierung gestalten.

Hierzu wird eine Lösung namens **Trust-Based Access Control (TBAC)** vorgestellt. Angewendet insbesondere auf den Bildungs- und Wirtschaftssektor werden fundamentale Vertrauensfragen in dynamischen föderierten Umgebungen gelöst.

Anforderungen an die TBAC-Lösung werden mittels dreier Basisszenarios, die die verschiedenen Klassen von so genannten Circle of Trust (CoT) in föderierten Umgebungen beleuchten, gesammelt. Durch Gewichten und Klassifizieren dieser Anforderungen ergibt sich ein Kriterienkatalog. Dieser dient als Basisreferenz für den Lösungsansatz. Zusätzlich wird vorhandene Literatur zum Thema "Vertrauens-Management" anhand dieses Kriterienkataloges analysiert und bewertet.

Die Analyse und Konzeption eines Vertrauensprozessmodells (Trust Process Model) dient der Beseitigung von Defiziten existierender Mechanismen. Es dient weiterhin als Rahmenwerk zur Einrichtung eines CoT unter den Organisationen, so dass deren sichere und vertrauensvolle Zusammenarbeit gewährleistet ist. Das Vertrauensprozessmodell spezifiziert die Entwicklungskette der Vertrauensbeziehungen durch die Phasen Initialisierung, Management, Validierung, Entwicklung und abschließender Prüfung.

Diese Phasen werden anschließend in dem TBAC-Rahmenwerk umgesetzt. Ergebnis hiervon ist die Spezifikation eines allgemein gültigen Satzes logischer Methoden und Prozeduren für die Beurteilung des Vertrauens hinsichtlich verschiedener Bezugspunkte. Zugleich kann dies als Basis für eine Implementierungsarchitektur dienen. Unterschieden wird primär zwischen mindestens zwei Klassen von Vertrauensbeziehungen: Vertrauen der Zusammenarbeit (Collaboration Trust) und der Qualitätsinhalte (Content Quality Trust). Die Vertrauensbeziehungen entwickeln sich dann aus den Erfahrungen bisheriger Zusammenarbeit unter Berücksichtigung zusätzlicher Faktoren und Verhaltensregeln wie z.B. Dienstgüte-Parameter.

Abschließend erfolgt die Analyse der prototypischen Implementierung des TBAC-Rahmenwerks und der hier entwickelten Algorithmen zur Berechnung von Vertrauenswerten. Dabei werden Kriterien wie Schnelligkeit, Genauigkeit, Wahl geeigneter Bewertungsmetriken sowie weitere Parameter zur Leistungsbewertung des vorgeschlagenen Lösungskonzepts herangezogen.

Contents

1	Introduction	1
1.1	Overview	1
1.2	Challenges	3
1.3	Motivation and objectives	4
1.3.1	Conception of a process model	5
1.3.2	TBAC Framework	8
1.4	Outline of the thesis	9
2	Requirements Analysis	13
2.1	Definition of Terms	14
2.1.1	Federated Environments	15
2.1.2	Technical definition of the CoT	19
2.1.3	Classes of CoT	29
2.2	Circles of Trust Scenarios	34
2.2.1	Scenario 1: CoT in academia field - IntegraTUM Project	34
2.2.2	Scenario 2: Dynamic CoT - Multimedia Digital Library Case Study	47
2.2.3	Scenario 3: Virtual CoT - DEISA Grid Project	65
2.2.4	Conclusion: Need of a generic model of CoT	73
2.3	Use Cases for the management of CoT	73
2.3.1	Requirements for the extension of the CoT with a change management process	73
2.4	Assessment of the requirements	74
2.4.1	Classification and weighting of the requirements	74
2.4.2	Summarization - Criteria catalogue	84
3	Related Works in Trust Management and Access Control	87
3.1	Trust definitions	89
3.1.1	Trust establishment and trust relationships	89

3.1.2	Circle of Trust (Liberty Alliance Project)	96
3.2	Indirect trust dimensions	100
3.2.1	Indirect trust by delegation	100
3.2.2	Indirect trust from past experience	101
3.2.3	Indirect trust by reputation	104
3.2.4	Indirect trust aggregation	106
3.2.5	Fulfillment of the requirements?	107
3.3	Interorganizational access control mechanisms	107
3.3.1	Intraorganizational access control models	108
3.3.2	Extension tentatives to interorganizational scenarios	109
3.3.3	Shortcomings and fulfillment of the requirements	109
3.4	Policy control	110
3.4.1	Privacy management	110
3.4.2	Risk management	111
3.5	Organizational Trust	112
3.5.1	Defining Trust by law	112
3.5.2	Discussion	113
3.6	Content quality trust	113
3.6.1	Wikipedia Case Study	114
3.6.2	Shortcomings and fulfillment of the requirements	116
3.7	Prototypes – Solutions for automated trust assessment	116
3.7.1	PolicyMaker and KeyNote	117
3.7.2	Trust Policy Language (TPL)	117
3.7.3	REFEREE Trust Management Model	118
3.7.4	Standards for the World Wide Web	118
3.7.5	Shortcomings of these automated trust assessment systems	119
3.8	Analysis and conclusions	120
3.8.1	Discussions	120
3.8.2	Update of the criteria catalogue	121
4	Trust Process Model	125
4.1	Conception of the trust process model	127
4.2	Phase 1: Initialization	128
4.2.1	Modeling Trust	128
4.2.2	Trust Assessment	136
4.2.3	Content Quality Trust and QoS Trust	159

4.2.4	Aggregation between the three dimensions of collaboration trust	162
4.3	Phase 2: Storage and management	165
4.3.1	Organizational models	165
4.3.2	Data structures	167
4.3.3	Risk managements aspects	169
4.4	Phase 3: Validation	170
4.4.1	Establishment of Trust Agreements	171
4.4.2	Policy Control	173
4.5	Phase 4: Evolution	177
4.5.1	Monitoring	179
4.5.2	Assessment and evaluation of the monitoring information	179
4.6	Phase 5: Auditing and Change Management	180
4.7	Evaluation and conclusion	180
5	Trust-Based Access Control Framework	183
5.1	Conception of the TBAC Framework	185
5.2	Trust Broker	188
5.2.1	initializePackage	188
5.2.2	searchPackage	193
5.2.3	storagePackage	201
5.2.4	aggregatePackage	205
5.3	Storage Components	208
5.3.1	Trust Agreements Repository	210
5.3.2	Resource Description	215
5.3.3	Auditing the interactions	218
5.3.4	Identity Repository	220
5.4	Access Decision Engine (ADE)	221
5.4.1	Access decision policies	222
5.4.2	Privacy policies	223
5.5	Change Management	223
5.6	Summary and Conclusion	225
6	Evaluation and Performance Analysis	227
6.1	Structure and notations	228
6.2	Comprehensive real-world scenario: Federated Learning Environment	229
6.2.1	Principals' roles	229

6.2.2	Overall interactions and relationships among the principals . . .	231
6.2.3	Workflows between the three interaction types	233
6.2.4	Trust management issues and requirements	235
6.3	Applicability of the TBAC Framework	236
6.4	Performance analysis: What and how to evaluate?	248
6.4.1	Accuracy of the trust information	249
6.4.2	Trust Metric	253
6.4.3	Access Control	255
6.4.4	Integrability of the TBAC Framework	255
6.5	Discussions and Conclusions	257
7	General Conclusions	259
7.1	Summary of this thesis	260
7.2	Primary results and discussions	262
7.3	Evaluation of the criteria catalogue	264
7.4	Open issues and future work	264
	Appendix A	269
	Appendix B	275

Chapter 1

Introduction

*"It is impossible to go through life without trust:
That is to be imprisoned in the worst cell of all,
oneself."*

Graham Greene

Contents

1.1 Overview	1
1.2 Challenges	3
1.3 Motivation and objectives	4
1.3.1 Conception of a process model	5
1.3.2 TBAC Framework	8
1.4 Outline of the thesis	9

1.1 Overview

Security and privacy issues have long been investigated in the context of a single organization ensuring control over the users' access to resources. For protecting resources from unauthorized access, security policies are defined and managed statically within the boundary of an organization and are typically centrally controlled.

However, developing large-scale Internet-based application systems presents new challenges, as these IT systems are used increasingly to support the exchange of resources between users and service providers, such as the shared use of expensive computational resources by research laboratories, as an example.

Internet-based technologies, such as the Web technology and the emerging Web service technology enable people and organizations to share all types of resources in different application scenarios, such as Business-to-Consumer and Business-to-Business e-commerce, educational e-learning systems, and many other control and communication systems. These application areas typically involve a number of collaborating organizations sharing distributed and heterogeneous data, services, and other resources over the

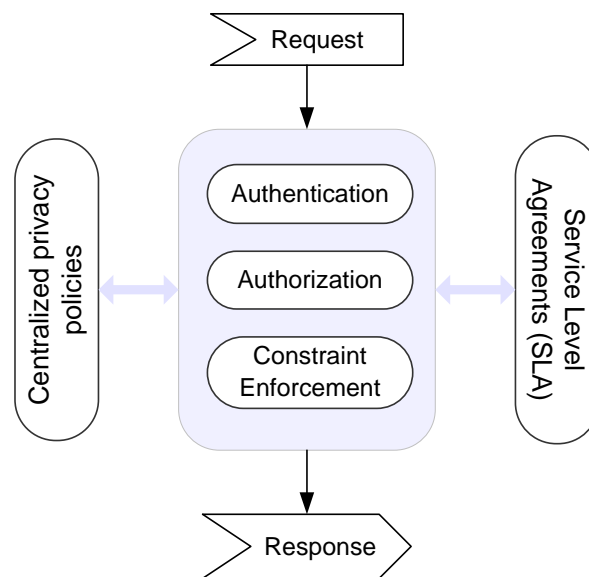


Figure 1.1: Traditional centralized access-control architecture within a single organization

Internet. The environment that aims at achieving resource sharing among collaborating organizations is referred to as a *Federated Environment (FE)*.

However, the increasing complexity of the distributed nature of resource sharing in FEs makes it difficult to control users' access since their identities are spread around several organizations; and this is because the users' identities authentication and access control of the resources are no longer appropriate to a single organization, but rather to the network of interconnected organizations. In this context, a primary high-level challenge relies on the interorganizational access control to sensitive data and priced resources, so that new technical measures are required to support the participating organizations in granting permissions for external entities, while ensuring the continuity of their underlying business processes.

In the context of protecting the resources within a single organization, users' identity-based authentication and role-based access control for authorization, which are usually enforced by the organization's security and privacy policies as well as other constraints, as shown in Figure 1.1, proved to be very effective in governing the way resources may be disclosed. From the organizational point of view, the Service Level Agreements (SLAs) within an organization, or between organizations in the FEs, usually characterize privacy constraints in line with structured sets of attributes and associated metrics (quantitative or qualitative) for the different facets of the privacy aspects.

Various access control models have successfully been applied to adopt these security and privacy techniques in intraorganizational scenarios and have later been extended for interorganizational and federation scenarios. Several variants of standards like Role Based Access Control (RBAC) and its successors, e.g. Attribute Based Access Control (ABAC), allow the delegation of administration on the one hand and privileges on the other hand. These access control solutions typically contain information about the objects that have to be protected (e.g. data files) and the subjects (e.g. users) which have the right to access these objects; however, the management of access control by these

systems requires that the objects have knowledge about all the potential subjects, which might access the objects. Unfortunately, they are only seemingly a good starting point for the inclusion of external entities in FEs, because privileges may only be delegated to those principals which are known beforehand in the federation.

Obviously, in order to make it possible for an authenticated user to be recognized and take part in personalized services across multiple domains, new mechanisms need to be investigated because the traditional security mechanisms are tightly coupled to the organization infrastructure and are not adequately flexible for dynamic changes.

Based on that, there is an evident need for specifying and enforcing the agreements established by collaborating organizations with respect to *trust* and security issues. These trust agreements are needed to establish interorganizational *trust relationships*, and thus, by defining the question of how trustworthy the external user is, service providers and resource owners may gain more knowledge and confidence about granting resource usage permissions.

1.2 Challenges

Assessing trust digitally has become a widely known research field in the last few years. The intended field of applications comprises trust in very different areas; examples are managing access control by using trust, trust for collaboration in virtual organizations and communities, estimation of trustworthiness of information and users in web-based communication platforms, trust for electronic commerce and others.

In such applications, most of the answers to trust relationships issues often depend on whether the entity we want to communicate with, is someone inside or outside the organization. Other questions relate to cooperation's agreements and associated authentication and communication protocols, as many of the challenges in FE come from the demand to grant single sign-on access to a collection of resources that might well have different, even contradictory, access-protection rules, and thus considering the human factors beside those technical aspects is more and more becoming a crucial question.

We delineate the well-known characteristics of trust in federated environments and their corresponding research challenges as follows:

- **Trust agreements:** Collaborating organizations in FE typically have their own security policies to enforce organizational security and privacy constraints. In such policies, the collaborating organizations typically negotiate issues, such as what resources can be shared, what rules should be enforced to authenticate and authorize legitimate users in the FE, as well as other technical issues such as the protocols that should be employed to securely exchange information and resources.

However, authorizing external users who may try to access the resources needs to be dynamic and content-triggered; because it is simply not possible to predict who may need to access the resource, and therefore manage users' accounts for these situations. Consequently, there is an increasing need for establishing interorganizational *trust agreements* and *trust policies*, which enable users, service providers and resource owners to express and enforce the trust they have in others.

- **Access control:** The enforcement of these trust agreements and policies calls for a trust-based authorization infrastructure that allows negotiated access to resources. We refer to this access authorization as *Trust Based Access Control TBAC*. Note that the newly established trust agreements and rules should not conflict with existing organizational policies and constraints. Although, their enforcement mechanisms are different from the security enforcement mechanisms at the infrastructure level, they can still make use of the existing security infrastructure.
- **Dynamic aspects:** Usually organizations collaborate in FE for the purpose of achieving resource sharing, frequently in a statically coupled manner. Due to the distributed nature of the FE, that collaboration may be short-lived and may change over time, i.e. service providers and groups of users (internal entities) may enter and leave the federation as their roles and responsibilities change.

Furthermore, trusting external entities in the federation typically depends on the internal entities, who may vouch or strictly forbid access privileges to these new involved entities. Therefore, having such high dynamics and fluctuation regarding the internal entities, entering or leaving the federation, it is hard to keep their vouchings up to date and thus relevant in order to predetermine the external entities' privileges.

On the one hand, organizations need to set up trust relationships and agreements quickly and efficiently to maximize productivity and eliminate the manual processes that often take place nowadays. However, on the other hand, these trust-based agreements need to be *dynamically* adjustable as changes occur as well as misuse and unauthorized access might be unforeseeable by out of date vouchings.

- **Risk aspects:** Sharing mainframe resources across organization boundaries in FE can drive innovations and lower the cost of cooperations. This typically implies that communication between collaborating organizations may go through multiple intermediaries rather than direct communication within a single domain. For this reason, automatically granting access permissions to previously unknown users, based on intermediary warrantors at multiple network sites, induces applications in collaborating organizations to be exposed to a higher risk of security threats; because the degree of trust in these intermediaries might be subject of verification as well. In this regard, another important aspect of building trust is also the *risk*, for which trust must be balanced to.

1.3 Motivation and objectives

The purpose of this research is to investigate issues on trust management, as introduced in section 1.2, and conduct accordingly basic solutions that support Internet-based collaborative and federated environments to manage the trust relationships with the TBAC Model. However, the realization and the development of this solution involves diverse requirements such as enabling organizations to integrate the TBAC model within their existing security technologies as well as providing the alternative of using a platform-independent TBAC decision point. This comprises an effective end-to-end trust model between organizations from anywhere in the set of domains in the FE.

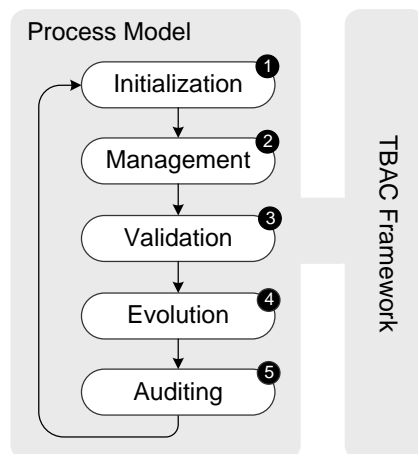


Figure 1.2: Relationships among research objectives

Concretely, this study will conduct basic research through two specific accomplishments: First, we will introduce a **Process Model** in order to define and specify the *Evolution Chain* for establishing trust relationships. That is, this process model, composed of five phases, begins with the *Initialization Phase*, where the trust agreements including Interorganizational security policies and constraints regarding message exchanges and resource sharing need to be collected and initialized.

The remaining phases attempt to establish shared meanings for formalizing new interorganizational trust relationships, so that they can be managed within the FE (i.e. classified and archived); validated on a set of criteria for a conventional access control model; evaluated in run-time and audited. These different phases will be detailed in the subsequent section 1.3.1.

In the second stage of our research, we develop a **TBAC-Framework** for a federated environment to demonstrate the enforcement of the Interorganizational access policies and constraints through our process model. This framework draws extensively of criteria from the process model as well as from the available used access control mechanisms in the FE. However, the framework is not an access control software as such – or even a collection of access control tools of different organizations in the FE – but rather a conglomeration of various technologies and activities designed to facilitate and promote the trust level assessment process and the effective building access control decision point. More details are given in section 1.3.2.

The two specific objectives of this research are described below. Their relationships are shown in Figure 1.2.

1.3.1 Conception of a process model

A typical trust-based transaction in a resource sharing FE will be as follows: An unknown entity will query for a particular resource using the available communication protocol in the FE. The requested service provider, who is offering the resource and might receive such requests from various groups of entities, will need to identify the unknown entity and get more information about how to trust him (such an information

is usually characterized by a *trust value*).

Further, after interacting with the entity based on the trust information, it will additionally be necessary to rate the resource or the service usage by the entity, for instance based upon its performance and vice versa. Five important issues are involved in this process:

1.3.1.1 Initialization Phase

In situations, where the requesting entities come from outside the FE, many existing FE systems tend to use community-based reputations to estimate the trustworthiness and predict the future behavior of these entities. In doing so, the current systems need to associate with each entity a trustworthiness metric and allow other entities to have access to this information and decide by themselves whether to interact with that entity or not.

In this `Initialization` phase we will investigate which search as well as which trust computation mechanisms can be effectively used in the FE in order to identify unknown entities, and correspondingly affiliate *trust level values* to their identities. This investigation includes also mechanisms for determining how metrics for such trust level values can be uniformly defined and based on entity reviews; so that each entity, when interacting with another entity, rates the performance of the entity on a common scale and vice versa.

The computed trust values can be helpful to state, for example, that a high trust value indicates the entity has gained good reputation in terms of its past performance and thus is more trustworthy, whereas a low trust value means the entity has relatively poor performance in the past and is rated with low reputations by other entities in the community. Such trust values, usually represented in trust relationships, and their potentially complex behavior are not yet fully understood.

In this thesis, the analysis of this understanding will be based on the different meanings and concepts of Trust and Reputation Management. However, looking to the distributed nature of the FE, the trust values can result from different trust metrics (internally defined in single organizations). In this respect, mechanisms for aggregating the prospective resulting trust values need to be explored as well.

1.3.1.2 Management Phase

Whereas the first phase is related to issues such as trust search and computation models used for building trust relationships among entities (either inside or outside the FE), the `Management Phase` is more related to issues for distributing, storing, and accessing the computed trust values among the involved entities securely. Concretely, this involves investigations on distributed storage of trust values in different locations in the FE, as well as secure access to these values against possible misuses and abuses by malicious peers.

As we will discuss in Chapter 3, a fair amount of work has been done in the area of trust level computation; however, the area of developing secure underlying mechanisms to distribute and access the trust values and ratings in distributed environments is relatively unexplored.

1.3.1.3 Validation Phase

The primary objective of this phase is to help assure that the development of the trust quantification and the management process from the previous phases results in a trust-based access control model that will perform as intended. That is, by predicting the unknown entity's trustworthiness, the TBAC model will support resource access decision taking by the requested organization or service provider. On the other hand, we will also define the influence and the impact of interorganizational security policies and constraints that govern the interaction, coordination and resource sharing of collaborating organizations.

However, as previously discussed, the decision of whether to grant or deny access does not only depend on the trust level just formed about the requester. Just as essential are the risks involved in the interaction as well as the *Quality of Service (QoS)* requirements. Therefore, our research aim is to explore how risk management on the service provider side and user trust level management frameworks can be combined and applied to trust-based access control mechanisms.

1.3.1.4 Evolution Phase

In the previous phases, we introduced issues for storing, accessing and reasoning about the trust value of the unknown entity regarding resource access, because this solution is intended to prevent entities to hold their own trust values (in which case, every requester would pretend to be the most trustworthy).

Apart from this, an additional issue relates to the need of keeping such trust values up-to-date, by investigating for example ways to recover from a bad reputation when freshly obtained trust information reflects a considerable increase in the confidence. That is, assuming the interaction took place, feedback about the requester's trustworthiness, as perceived by the requested service provider at a given point of time, can serve as an input parameter in the trust evolution process, whose goal is to achieve a run-time reevaluation of the trust relationships, and thus keep them accurate.

Obviously, the incorporation of the time dimension as well as the monitoring aspects (for testing and tracking how the new entities use the granted service, either for a short or a long run of the service access) to prevent access decision from capitalizing excessively on past interactions is very crucial in the context of trust management and will be, thus, part of this work.

1.3.1.5 Auditing Phase

The last phase of our process model will be characterized in the *Auditing Phase*, which closely follows the *Evolution Phase* to ensure a certain level of quality for establishing and storing the desired trust relationships and to ensure that this level of quality is maintained consistent through run-time changes of the entities' behaviors and interactions in the FE.

To achieve this, an auditing evaluation of the previous phases will be performed both quantitatively and qualitatively. The quantitative evaluation focuses on the technical performance views that auditing introduces in each phase, while the qualitative evaluation complements the quantitative evaluation by introducing additional human factors

such as self-established access control constraints and conditions.

On the basis of this evaluation and in order to minimize the number and impact of eventual related security incidents upon the released services and resources, an efficient *change management* process might be promoted to take these changes into account.

1.3.2 TBAC Framework

After having discussed the process model we will now concentrate on the conception and the design of the final framework of the trust model. This research activity focuses mainly on the development of mentioned methodologies and automated reasoning tools, including privacy and risk management that can be used for verifying and assessing the relevance or limitations of the theoretical models and to create bridges toward large-scale applications.

The process model, discussed in section 1.3.1, will be realized within a TBAC framework, which represents the second part of our research. This framework will be designed to be simple and extensible at the same time in order to define standard modules that interact with a variety of sources of information in the FE for making trust management decisions. For the objective of modeling constructs for interorganizational trust and security, it will encompass two major components that carry specific responsibilities:

- **Trust Broker:** This component will first collect relevant information about the requester for computing the prospective trust level by means of trust level computation algorithms and aggregation mechanisms in terms of mapping functions for Interorganizational trust level schemes.

Secondly, it specifies a unified definition of the shared resources in the FE, based on an XML-based trust agreement and resource description specification language, by which collaborative organizations can describe the different types of their resources and services (which are shared in the FEs and may be accessed by external entities), and can integrate their own access policies and constraints on these shared resources.

- **Access Decision Engine (ADE):** The resulting trust assessment as well as resource access rules will be integrated into an access decision engine which processes the information collected from the trust broker and triggers a policy decision point (PDP). On the one hand, the PDP decides solely based on the provided information, which also includes the relevant access control policies and environmental information such as the current date and time. On the other hand, it preserves the autonomy of collaborating organizations in maintaining their access control over the resources they share.

The overall goal of our research is to design a TBAC Framework for distributed FEs that meets the trust establishment requirements and enables the members to form, update, and exchange trust levels of external users. To evaluate the effectiveness and the scalability of our TBAC Framework, we will present a prototype implementation to verify our research results. This prototype illustrates simulation-driven workflows, which translate high-level trust agreement specifications and data flow between the components, presented within the TBAC Framework, into events, action-oriented rules and

triggers. Based on that, this prototype will also consider issues for trust decision delegation and its automation in more complex scenarios, for example if the assumption is an invalid that a chain of intermediate entities exists which can be contacted on demand to acquire reputation information about the unknown entity.

Furthermore, it will be part of our work to evaluate the performance of our model with respect to the promptness by which reputation information is collected, the accuracy of the obtained trust judgments as well as the adaptability of the model to the collaborating organizations' distributed access control policies in the FE.

1.4 Outline of the thesis

In order to accomplish the goals outlined above, the organization of this dissertation, as detailed in the process model in Figure 1.3, is as follows:

In *Chapter 2* we will first present a broad definition of terms and discuss the technical meaning of trust in federated environments, the common used terminology as well as the aspects related to trust by considering pertaining human factors and organizational facets beside the technical aspects. Subsequently, we will sum up these definitions by introducing the perception of the *Circle of Trust (CoT)*, which will be formalized in terms of attributes and components. This concrete definition of CoT will be further enforced with three basic scenarios, which serve as examples in the presentation of our trust-based management approach. They will typify the different kinds of CoT, and will help to deduce the main requirements on such an approach. The chapter will then be concluded by a set of requirements that will be collected and classified in a **criteria catalogue**, which in turn will serve as a basic reference for the following chapters.

Chapter 3 presents a comprehensive survey of the literature that can be found on trust and reputation management in distributed and federated environments. It will primarily review a number of approaches that have been done for the fulfillment of the trust requirements investigated in Chapter 2; and accordingly, illustrate the deficiencies and the weaknesses that still need to be resolved. On the basis of this analysis, we will explain the differences between our work and other existing research projects, and point out our contributions in coherence with the given criteria catalogue.

The design and the realization of the different phases of the process model, introduced in subsection 1.3.1, will be processed in *Chapter 4*. Basically, the contribution of this chapter is particularly relevant with regard to the **graph representations** of the collaborating entities in the CoT as well as the **trust computation algorithms**, which we will conceive in phase 1 for inferring and aggregating trust values among them. In the successive phases (for storing, managing and evaluating the prospective trust relationships), we will present some theoretical evidence of the accuracy of the algorithms and will show how these inferred trust values, when integrated into applications, can enhance the entity's experience.

Chapter 5 will leverage and enforce the theoretical aspects of the process model within a **TBAC Framework**, which supports experiments under different configurations; this is done by presenting the design and implementation measures of its two main components: (i) The distributed **Trust Broker**, so that individual entities can share their feedbacks about entities or services without the overhead of maintaining their own ratings, and (ii) the **Access Decision Engine** that reaches resource access decisions based

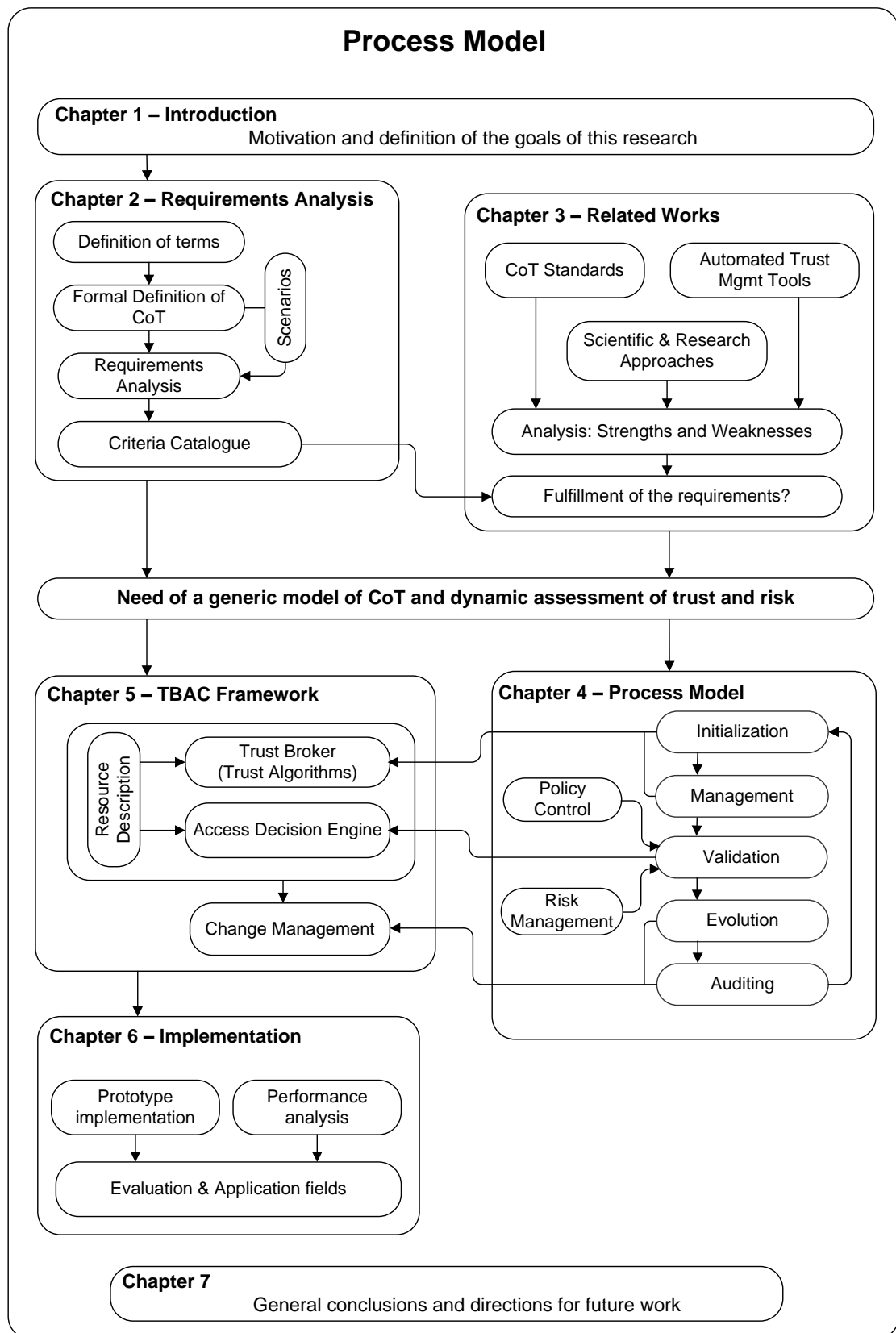


Figure 1.3: Process Model

on the trustworthiness degree of the requester entity.

In *Chapter 6*, we will analyze the prototype implementation of our TBAC Framework, and evaluate our algorithms in the light of performance criteria such promptness, accuracy, choice of the trust metric scales as well as several other performance parameters. In addition, we will conduct some simulation experiments by integrating our solution into the standardized trust management systems KeyNote (RFC-2704) to show some fields of application supported by our solution, and to envision the trust formation process in a qualitative as well as quantitative manner. In summary, *Chapter 7* will conclude the results of this thesis and will provide some insights into possible improvements and future work.

Chapter 2

Requirements Analysis

*"Whenever the people are well-informed, they can
be trusted with their own government."*

Thomas Jefferson

Contents

2.1	Definition of Terms	14
2.1.1	Federated Environments	15
2.1.2	Technical definition of the CoT	19
2.1.3	Classes of CoT	29
2.2	Circles of Trust Scenarios	34
2.2.1	Scenario 1: CoT in academia field - IntegraTUM Project	34
2.2.2	Scenario 2: Dynamic CoT - Multimedia Digital Library Case Study	47
2.2.3	Scenario 3: Virtual CoT - DEISA Grid Project	65
2.2.4	Conclusion: Need of a generic model of CoT	73
2.3	Use Cases for the management of CoT	73
2.3.1	Requirements for the extension of the CoT with a change management process	73
2.4	Assessment of the requirements	74
2.4.1	Classification and weighting of the requirements	74
2.4.2	Summarization - Criteria catalogue	84

The purpose of the requirement analysis, which we are presenting in this chapter, is to investigate requirements for trust management solutions in current approaches to federated environments and to analyze interorganizational trust aspects, which go beyond an isolated organizational model where each identifier that a user possesses can only be used for one isolated organization. Since disjointed organizations will have different trust requirements when they wish to cooperate between each other, as there are costs associated with establishing trust, it is necessary to define a unified criteria catalogue amongst them in order to coordinate the requirements on establishing the desired trust relationships. The realization of such cooperation agreements can have relatively complex and sometimes continuously conflicting trust requirements, and the end users have so far had little experience with them.

The contribution of this chapter aims mainly at analyzing the common trust requirements resulting from the various federated environments models and scenarios. Several definitions for the terms of *Circle of Trust (CoT)* in federated environments exist, but they all look at those terms in a different angle and are not yet standardized.

As illustrated in Figure 6.1, the dependencies between the Sections are illustrated in a process model. In Section 2.1 definitions, based on the technical criteria as well as on the organizational view for building CoTs, will be given. When building such CoTs, it is quite usual that the impact of static and dynamic aspects as well as additional specific aspects of virtualization on the subject play an important role. In Section 2.1 we correspondingly consider the existing definitions from the literature and classify the different types of CoT according to those aspects into three categories: *static*, *dynamic* and *virtual* trust communities.

Section 2.2 presents three example application scenarios that provide indications on the aspects and the definitions of CoTs in federated environments. They will illustrate the primary challenges and requirements associated with building and establishing trust relationships and will consider the role of those trust relationships among organizations in enabling successful outsourcing of personal data and services. The analysis of the given scenarios will show that the problems of trust management coming up there do not always fit into one single definition class and therefore cannot be achieved by deploying a one-sided disjunctive model. This analysis emphasizes also the need of designing a generic model of CoT that may be based on more than one model in order to help the entities fulfill inter-domain trust and privacy requirements. A requirements set for conceiving such a generic model will conclude this section.

In Section 2.3 other issues related to the governance and *Change Management* of the CoT (e.g. issues that arise with entities entering and leaving the CoT over a period of time, issues for dynamic update of trust relationships as well as issues for several CoTs overlapping between each other) will be discussed. Some use cases for each scenario, representing those changes and defining a goal-oriented set of interaction requirements between external actors and the CoT under consideration have been depicted and will be presented in this Section.

The needs and the requirements which are derived from each scenario and use case as well as from the existing paradigm of trust management will be then detailed in Section 2.4. At last an evaluation summary of the requirements in a kind of a criteria catalogue as well as some broader structural issues of virtualizing the CoT will close this chapter. This criteria catalogue will be broken down into 3 phases: (i) *Elicitation* (gathering, weighting and classifying the requirements), (ii) *Analysis* (checking for consistency, uniformity and completeness), and (iii) *Specification* of the criteria catalogue (writing down descriptive requirements and creating an initial bridge between requirements and solution design).

2.1 Definition of Terms

Several terms used for characterizing federations and trust in federated environments are not well-defined in general usage. To allow for a common understanding a definition of terms forms the beginning of this chapter. While Section 2.1.1 gives an informal definition to CoT, Section 2.1.2 provides a formal definition in order to evaluate the

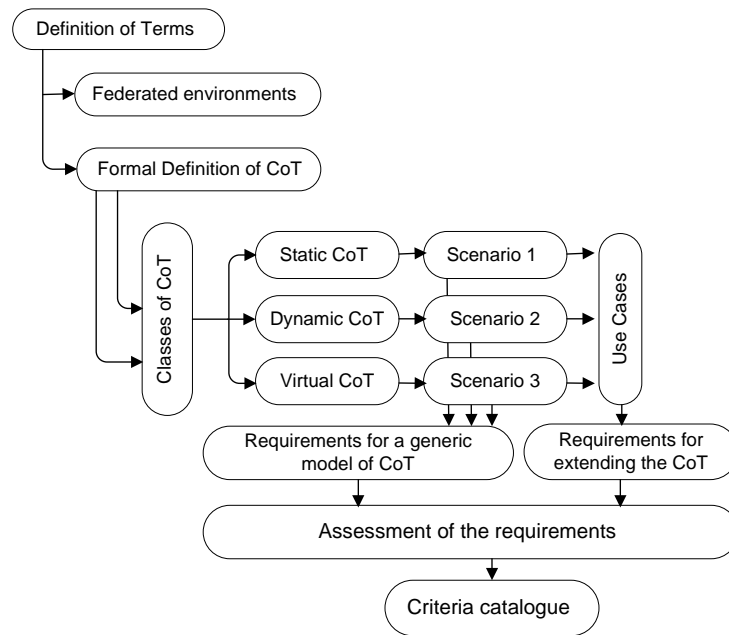


Figure 2.1: Sequence structure for chapter 2

CoT in light of considerations including components, attributes and characteristics. In addition, Section 2.1.3 will refer to three classes of the CoT, deduced from real-word examples. These definitions are valid throughout the whole thesis.

2.1.1 Federated Environments

In this thesis we have chosen trust in distributed federated environments (FEs) as a basic surrounding to study the larger issue of trust and reputation relationships, and afterwards we intend to apply this study in some very special areas such as in Federated Identity Management (FIM) and in other web-based applications. The decision to consider trust in federated environments is justified by the fact that they form a large, publicly available shared information space with tremendous interactions among the entities participating in the federation.

We define a FE as a collaborative environment between several organizations for handling resource sharing across multiple domains while protecting those same resources from unauthorized access. This definition is based on the formal agreement between organizations, and in this vein resource owners can determine how, when, and what resources are available for access. In government, an example application of this federated environment is the partitioning of business logic and data resources among different agencies often organized around central business models centers, known as Centers of Excellence (COE). Business operations or processes that are similar across agencies can be outsourced to one or more agencies that are specialized in that given business domain in such a way that a repeated processing of a business activity at a single agency can be avoided.

Another application example of a federated environment are FIM systems, where the

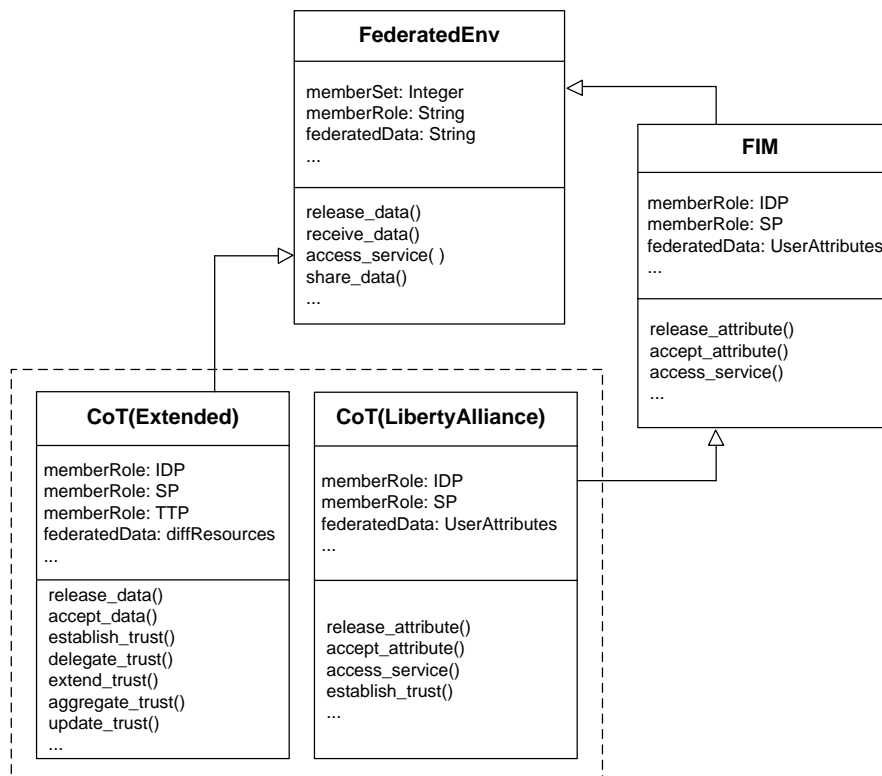


Figure 2.2: Federated Environments in conjunction with the CoT

partners couple their data repositories in order to facilitate collaborative and secure sharing of their user data and services. The partnering organizations will then have to outsource the authentication data of their users to their home organization, which is responsible for authenticating their respective users and vouching for their access to services. On the one hand this outsourcing allows individuals to use the same credentials at a single point to sign on to the federation of more than one organization, and on the other hand it allows organizations to share applications without the need to adopt the same technologies for directory services and authentication mechanisms. Under those circumstances, each organization must trust its partners to vouch for their users and services, thus building a trust community, known as a **Circle of Trust (CoT)**.

In the context of FIM systems, the CoT is characterized as a federation of identity and service providers whose purpose is to facilitate business relationships with regard to security and privacy concerns. The term of Circle of Trust in the Liberty Alliance Project (LAP)¹ is defined as: *"group of service providers and identity providers that have business relationships based on Liberty architecture and operational agreements and with whom users can transact business in a secure and apparently seamless environment"* [VSH05].

However, the purpose of the CoT in LAP is to deploy a Liberty-enabled architecture of identity management specifications and technology by assisting stakeholders and their FIM partners in identifying the legal structure best suited for their cooperations. Such structures and contractual agreements among participating parties serve to create

¹<http://www.projectliberty.org>

a trusted and legally binding relationship among the participants.

In contrast to the LAP CoT frameworks, the purpose of our study is to analyze trust issues in more generalized federated environments, where federated data is not limited to the users' identity information and preferences, but additional other heterogeneous resources might be subject of federation as well, and where the access control policies are not necessarily enforced by the Liberty ID-FF Architecture [Was04].

Based on that in Figure 2.2 we present a diagram that shows some aspects of the inheritance of the class **FIM** from the class **FederatedEnv**. Although the type-like elements (for example the type-like of the organizations' roles or the type of the federated data type) and instance-like elements are not exactly the same, but they share many similarities.

Obviously, the class **CoT(LibertyAlliance)** inherits from the **FIM** class. The notation for doing so is simply enforced by the fact that the CoT framework is initially instantiated from the Liberty FIM project. With regard to trust, the **CoT(LibertyAlliance)** class accomplishes an extension of the super class by conceptually inserting additional action sequences regarding the establishment of trust. This extension is mainly based on the agreements between the partners inside the federation.

This motivates the investigation of a new CoT class, called **CoT(Extended)**, which allows to extend the LAP CoT into a more generic model that is intended to describe additional specific particular attributes, items and functions such as functions for trust delegation, extension, aggregation, etc. Therefore it is convenient to choose a notation for each type-instance pair of elements such that the correspondence is straightforward visually apparent.

Below we shall give some general definitions related to the fundamentals, benefits as well as the challenges faced in LAP CoT Frameworks. Subsequently, Subsection 2.1.2 will provide further information on the technical definition, notation and the relationships between the differing properties of the extended model of CoT.

2.1.1.1 Circle of Trust fundamentals

Circle of Trust (CoT) frameworks specify a common set of policies, procedures and collaboration interfaces within a group of organizations.

Instead of 1:1 relationships between principals, the CoT offers a sort of association where organizations can apply for membership. To become a CoT member, an organization is compelled to adhere to the specification, in particular to procure and operate prescribed software packages, and to demonstrate that CoT policies are respected and enforced. In return, the setup of cooperation with another CoT member organization is accelerated by the common base of interfaces and by an initial level of trust—the enrolment process supplies a form of *certification* of a fellow member.

Accordingly, *trust* in this context is built on a common set of rules, responsibilities, and commitments set forth in the *CoT Foundational Documents*. If two members of the CoT wish to cooperate, the trust foundation as well as the federation infrastructure are already in place. A technical definition of CoT will be given in Section 2.1.2. Furthermore, a detailed technical interpretation on how privacy and trust aspects are enhanced in federated environment will be illustrated in Section 2.2 according to the infrastructure and the techniques deployed in each scenario apart.

2.1.1.2 Benefits of membership

One of the benefits of the concept of CoT is the outsourcing capability. The participating providers within the CoT can save on a lot of aspects of their businesses, increase their profits and minimize the expenses for handling 1:1 relationships. By means of the operational agreements, SSO functionalities and an identity management infrastructure (exchange of authentication and authorization information), customers and other providers can transact business with any of the other providers in a continuous and secure manner.

As an *operative benefit*, this allows integration of the services facing the members' customers, while ensuring that user data is shared according to published policies. The CoT privacy policies (details about the different policies and rules in the CoT will be given in Subsection 2.1.2.6) require the CoT members to have the legal and practical ability to control and execute any outsourced functions to ensure that it has the ability to continue to provide transaction services without neglecting the concerns of the privacy-preserving user's data. However, the CoT ensures an *implicit initial level of trust* that can be exploited in reasoning about the trustworthiness of principals within and outside the CoT.

2.1.1.3 Circle of Trust challenges and issues

Taking into consideration the globalization of service provisioning together with shortened setup time until delivery (real-time/ad-hoc, at worst), current CoT specifications may be too rigid. The contractual framework together with a specification of duties for members render the application process slow, because new members have to wait until their submissions have been confirmed in order to be able to transact with other members. In addition, the benefits a CoT offers are only useful when *both* partners in a potential cooperation (e.g. to provide service to a traveling user's location) are members.

For a member, the CoT does effectively provide a trust base that can be leveraged in order to instantly estimate a trust value for a hitherto unknown potential cooperation partner. However, once such a CoT has been created, doors are opened to formal and across-the-board trust relationships. A problem may occur if for instance the user disclaims providing his information or might not be willing to entrust certain personal data to a certain online service just because it is a member in the CoT.

In this requirement analysis, we want to address the problem of trust management beyond the borders of the CoT and analyze to what extent external organizations and users can collaborate with the CoT members, either generally or for a given interaction. Essentially, we present the requirement of deriving trust assessment for entities outside the CoT. As a non-CoT organization may have business relations with some of the CoT members, a member can consult its CoT peers with regard to a non-CoT organization that requests cooperation. Their appraisal, formulated as a level of trust indication based on the requesting organization's conduct, can be employed as base for the initial level of trust for the cooperation.

2.1.2 Technical definition of the CoT

This section provides some semantic definitions of terms, key features and concepts, frequently used or referenced in the CoT. Usually formal CoTs have semantic definitions that are enforced by the federated environment where they are formed. CoTs in FIM systems, for example, are adapted to the terminology used in FIM Standards. Still other groups of CoT define terms in ways specific to their particular community. Most of these communities use or define these terms in slightly different ways, with old terms taking on new and varied meanings and new terms emerging. It is, therefore, often confusing to the CoT managers trying to communicate across the various domains. In this section, we recognize some common differences in well-known federated environments, and seek to provide generic definitions that we will use for classifying the CoT types presented in subsequent Section 2.1.3.

2.1.2.1 Trust dimensions

In the following, we first give some definitions to the meaning of trust, and provide further knowledge on some of its dimensions that are emerging in current communication mechanisms in federated environments.

Across organizations, trust typically develops between individuals who are embedded in a complex network of existing and potential relationships. In this thesis, we identify two different ways in which two entities may be linked to each other via the notion of trust. We distinguish between two classes of trust, (i) *direct trust* for entities that are known to each other through membership to the same domain, and (ii) *indirect trust* for those that are linked indirectly, for example via interactions with third parties, or transferably via recommendations from third parties.

Direct Trust:

As we see in Figure 2.3, direct trust can be built between entities (both inside the CoT) on the basis of the confidence, gained from the federation principles and the classical security tasks such as authentication and authorization. This type of trust, also denoted as *trust by membership*, entails that all entities enrolled in the organization will subsequently be provided with access rights and considered to be trustworthy. Trust is established, in this way, because it is possible to ensure that entities who attempt to perform actions in a system are in fact the entities who are authorized to do so.

Indirect Trust:

In contrast to direct trust, indirect trust addresses the case where the requester entity is outside the CoT (where access to the resource is requested), so that the two entities (the requester and the requested organization) do not have direct trust relationships (for example business agreements) with each other, but do have agreements with one or more intermediaries, which enables a trust path to be constructed between these two entities.

As we will discuss in Chapter 3, it has been shown by previous research that trust, in this regard, needs to be viewed as a multi-dimensional construct combining specific aspects and criteria. That is, because the trust path can either be conducted from a specific trust dimension or through an overall assessment of multiple trust relevant-dimensions.

To investigate how to construct such an indirect trust relationship from a multi-

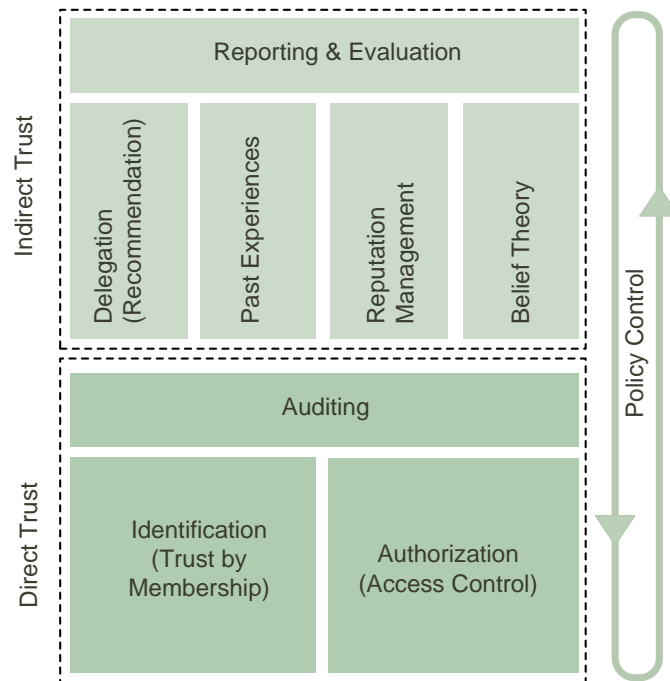


Figure 2.3: Trust definition - Direct and indirect trust

dimensional perspective, this study proposes some predominant dimensional scale of trustworthiness, as illustrated in Figure 2.3, dealing with delegation, past experience, reputation, and belief theory, and then shows the importance of examining the effects of each dimension individually.

- *Trust by delegation and recommendations*; one of the well-known trust dimensions is the so-called delegation system which enables entities to express and enforce the trust they have in others. The basic idea behind delegation is that a known entity in a system delegates authority to another unknown entity in order to carry out some functions.
- *Trust from past experience*; distributed auditing systems allow entities behavior to be continuously monitored during ongoing two-party exchange. This vision of trust is suited to reason about future interactions on the basis of the outcome of past ones, because it performs a continuous monitoring of interactions as they take place.
- *Trust by reputation*; the concept of sharing reputations of entities in online communities is to provide ways for maintaining trust in these communities. This is achieved by the provision of information about the entities' past performance, by collecting, distributing and aggregating feedback about past behavior.

Reputation management systems significantly increase trust, especially in eCommerce scenarios, and thus, the volume of trade. Because attributing positive reputations usually encourages the recipient of that reputation to trust more, and attributing negative reputations may work as a sanctioning mechanism to punish

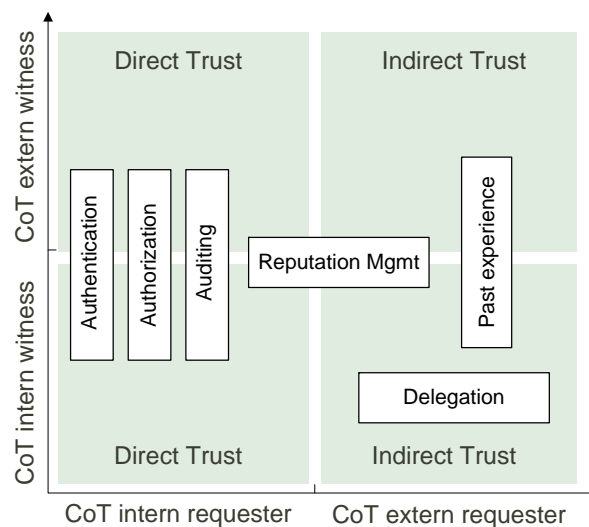


Figure 2.4: Affiliation of the trust dimensions in connection with the position of the requester and the witness with regard to the CoT

dishonest behavior, which makes the owner of this reputation behave in a more trustworthy way.

- *Trust by belief*; the aspect of belief indicates an additional dimension of trust, corresponding to the case when all of the above defined dimensions of trust are missing (i.e. the external entity is totally unknown to the CoT). Trust, in this vein, can be estimated by means of a theory and an expectation about the kind of motivations the unknown entity is endowed with, and about the question what the prevailing motivations in case of conflict will be.

The multitude of trust dimensions and their corresponding estimation methods give rise to additional questions. Most importantly, it is substantial to investigate selection criteria for invoking the appropriate method, the one that is requested for most. For this issue, there are a number of parameters that need to be considered with care; namely, the position of the requester as well as the witness in relationship with the CoT is of relevant importance.

Actually, as illustrated in Figure 2.4, direct trust may always apply when the requester is someone inside the CoT, because with the help of his identity information and origin, it can be ascertained what functions he is allowed to carry out.

In the opposite case, i.e. the requester is someone outside the CoT, the position of the witness will then be decisive. This is because, one can opt for the delegation techniques, for example, only if the identity of the witness is prior assigned in the CoT. Note that the trust by reputation dimension can be applied in all cases, because it may always be helpful to enforce the notion of trust even if the requester is someone known in the CoT.

Each of these dimensions is argued to influence the process as well as the quality of trust establishment among two distinct entities. Figure 2.5 illustrates a classification of the different identification modes in the context of organizations within the CoT having to deal with external entities.

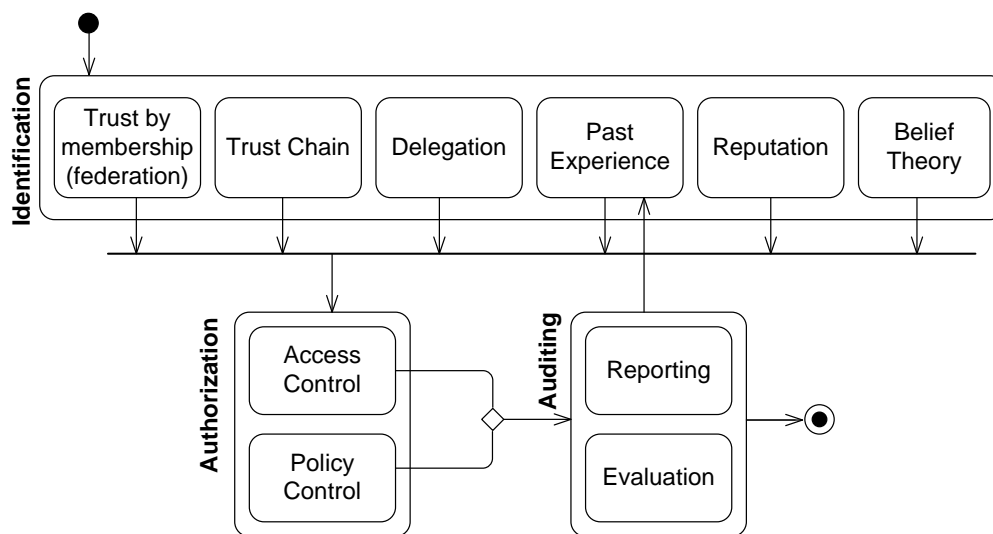


Figure 2.5: Classification of trust dimensions for access control

In order of priority, we see that trust by membership (direct trust) has to be performed foremost when the requester is someone inside the CoT. In the opposite case, the trust relationship can then be appraised from the different trust dimensions depending on the information collected about the external entity as well as the rules used for aggregating this information.

2.1.2.2 Principals

A group of previously unrelated and autonomous entities (either persons or organizations) form the CoT, with the shared purpose of protecting target resources, and keep a certain level of trust among the entities. Accordingly, we refer to the *Principals* as the entities who are participating in the given CoT, i.e. either entities whose identity can be authenticated and federated, provider entities who provide and share services, or entities that are responsible for the management or the technical implementation of the CoT platforms. Note that entities that are outside the CoT and cooperate with members of the CoT are likewise provided with the role of principals. Principals' roles will be detailed in Subsection 2.1.2.5.

In the CoT where resources are shared, the main kinds of principals need to be connected by trust relationships, and therefore need to be identified, often by means of their particular identity lifetimes: People can make assertions affiliated to their long-lived IDs; and organizations, which are generally a set of people and computers, can make close evidence of their identity from their DNS, IP-Addresses, or any other standard identifier. The *credentials* describe each kind of principal, often as attributes attached to that principal and its relationships to other principals (other informational attributes of the principal, such as its name and birth date, are not relevant for trust building purposes and will not be used in the context of this thesis).

We consider any characteristic element which is used for trust identification purposes as

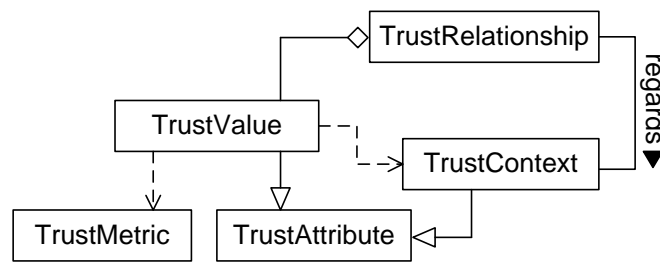


Figure 2.6: Design of trust relationships

an identifier or a credential of the identity. We assume that identities within one single CoT are unique and injective, i.e. no two human beings or organizations may have the same identity. The set of the trust credentials form the *trust profile* of the principal.

2.1.2.3 Trust Relationships

Trust is often built through the actions and interactions one principal has with one principal and other principals in the CoT. All of these actions and behaviors can have a short and a long-term effect on the other participants with whom the principal surrounds himself, and that is, establish *trust relationships* with them. In this context, the trust relationships give the principals a formal way of expressing their confidence in other principals as well as a way of checking which principals have expressed their confidence in others. Correspondingly, this results in a structure of trust-relationships formed between all principals in the CoT.

However, the same person or the same organization can have different trust relationships in different *trust contexts* — the context refers, in general, to the interaction that happens between two principals for a specific service usage or resource sharing —, and therefore each principal’s identity is reflected by a different set of trust relationships. The appraisal of the outgoing interaction can have a meaningful trust-enhancing or a trust-diminishing impact on the strength of the relationship. Figure 2.6 illustrates some obvious interdependencies and ties for a trust relationship.

Different types of trust relationships can be quite varied in their characteristics, and may be ephemeral or permanent; applied or inherent as it is the case in PKI systems; self-selected by individuals in the community or issued by an external authority, such as deduction from a social network; interpretable by humans, or automatically assessed, or both, etc.

In most known federated environments the principals use a community-based scale, usually characterized by *trust values*, to estimate the trustworthiness and predict the future behavior of principals. Concretely, the current systems associate a trust value with each trust relationship, which in turn is classified according to a *trust metric*; and by allowing other principals to have access to this information, they can decide by themselves whether to interact with unknown principals or not. In the following section, we introduce some broader aspects for defining trust metrics and associating them with trust relationships.

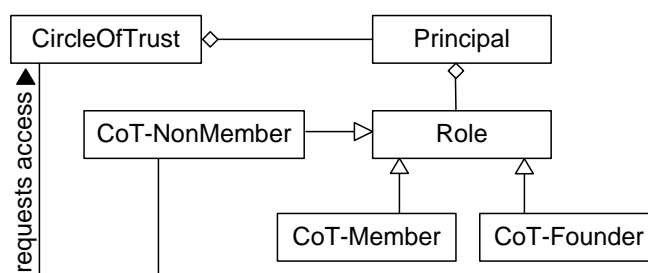


Figure 2.8: Roles of the principals in relationship with the CoT

2.1.2.5 Roles of the principals

The roles of the principals we identify in the CoT are those of CoT members (providers, organizations and users involved), CoT non members (external principals who tend to cooperate with the CoT members), and CoT administration and founder organizations. Mapping the entities to roles will also involve to define requirements and obtain knowledge in different ways about the access rights, as the success of the management of CoT requires the right roles to be enacted in order to achieve a specification of the grouping of tasks a principal can perform.

We recall the illustration of the trust definitions in Figure 2.7 and complement the *principal class* with an additional *role class* in Figure 2.8 in order to sketch the different roles of the principal in relation with the CoT as follows:

CoT members, the primary role in the CoT is defined in *CoT-members*, which generally constitute a group of independently owned enterprises – and other organizations – that provide each other member with access to their underlying shared resources. The CoT-members as service providers may be directly the owners of the resource, and thus can dictate the terms and conditions under which that resource can be used, or may have been assigned the right to represent resources of other service providers either inside or outside the CoT. The Member group bearing this seal are usually carefully selected as having an acceptable customer service rating, a good reputation for distributing quality services, and a commitment to respect the membership rules in the CoT.

In this thesis, we will investigate the way the fulfillment or the violation of such conditions and rules can be represented in a form of a trust level (we defined the trust levels in Subsection 2.1.2.3), and consequently the extend to which the membership can accordingly be influenced.

In FIM, all the entities inside the federation are viewed as members, for which three main roles are defined in the identity federation protocols and supported by several implementations: (i) Identity Provider (IDP) which is responsible for managing, authenticating, and asserting a set of identities within a given circle of trust, (ii) the end users representing these identities, and (iii) Service Provider (SP), which affiliate with the IDP by providing services to the principals while relying on the IDP to authenticate the principal's identity correctly.

CoT non members, this role is referred to as all the principals outside the CoT seeking to become a member of the CoT or to cooperate individually with one of the CoT members. This broad category includes practically any organization on the Web today,

for example Internet portals, financial or educational institutions, government agencies or any other non-profit-making organizations. From this perspective, a users' group from which requests originate (i.e. users requesting permission to access resources on sites residing inside the CoT) typify the role of *CoT non members* as well.

CoT founders, the founders of the CoT are the governing entities that have the role to collectively lead the entire CoT and handle regiment issues while also having the possibility to act as a normal CoT member, for example to participate in resource federations. The role of the founders comprises duties such as approving the CoT members which bear the seal, revoking the member's seal when the member is considered to violate the law, e.g. in cases the terms of the agreement are broken or compromised in any way by the member (see the rules in Subsection 2.1.2.6), and it comprises the general administration and hosting the board of the CoT in order to make sure events and interactions are planned and take place.

It is noteworthy that every CoT-Member can be the owner of the resource or simply a trusted third-party that can be solicited from another CoT-Member to get more knowledge about the external user. CoT-Founder can act as trusted third party as well.

2.1.2.6 Operational rules and agreements

As denoted in the previous definitions, the Circle of Trust is an evolving community ambition, whose duration, stability and quality of cooperations between the CoT-members strongly depends on the community effort. Some measures like specifying quality of service (QoS) parameters and penalties for not fulfilling them are, nowadays, integrated within a number of classical approaches to mitigate several of the customer's and the service provider's trust values resulting from interorganizational dependencies and business connections.

The necessity of service level management (SLM) and its interfaces to other IT service management (ITSM) processes, especially financial and security management, have been motivated, analyzed, and improved by both researchers and practitioners over the past decades and it is impossible to imagine having to do without them. By means of the service level agreements, which are formally negotiated between the members, the resource owners in the CoT will have the possibility to determine how, when, and which resources are available for which kind of access by whom.

Usually, granting permissions to a customer's users, reflects that each of these users is sufficiently trusted and that the risk of incidents caused by the users is outweighed by the mutual benefits. Similarly on the provider side, in order to establish trustworthiness, service providers must supply the reliable, and consistent service levels that SLAs have promised, and they must be able to prove it.

By definition, the SLA records the common understanding within the CoT about requirements of using services and specifies certain levels of performance (such as serviceability, throughput, or availability). However other attributes of the service like billing, penalties in the case of violation of the SLA and even outsourcing SLA relationships must be meticulously identified among the CoT members. Rather than designating these agreements properties as SLAs' attributes, they relate more closely to the trust relationships; therefore throughout the thesis we will focus solely on these SLA-trust-specific-attributes, which will be referred to as the *Trust Level Agreements (TLA)*.

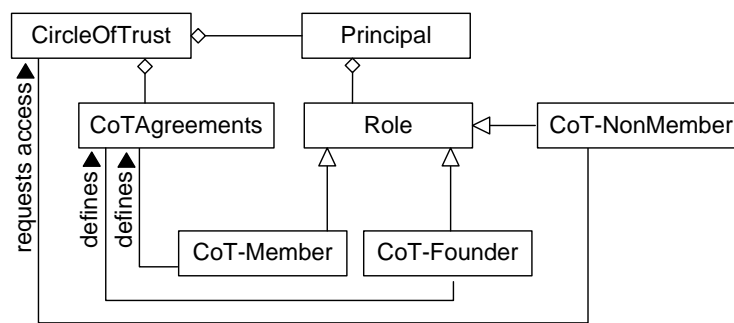


Figure 2.9: Agreements Rules in relationship with the CoT definitions

As can be viewed in Figure 2.9, both members and founders of CoT have separate ways to set up the TLA rules, depending on the services they are representing as well as on their roles in the CoT.

With the underlying principle understood, a determining factor for creating the CoT is to gather business requirements, usually expressed with TLAs. These requirements must be specific to the organization, although they may well be similar to those of other comparable organizations. In the following, we give some exemplary cases of how requirements can be determined by both the founders and the members:

Founders; a number of important duties are assigned to the founders of the CoT represented mainly in the way in which CoTs are expected to deal with the principals' memberships. These duties are set out in a form of rules such as defining:

- rules of granting or canceling the principals' memberships. For example, if a member decides to dissolve his membership with the CoT, rules for relinquishing any or all rights to past or future benefits resulting from the CoT need to be pre-defined and accordingly deployed by the founders; including any consequences that may occur or measures that have to be taken, such as archiving or definitively removing the member's data from any CoT platform and domain (we will refer to the CoT platform in Subsection 2.1.2.8), when the membership is dissolved.
- rules and conditions for revoking the membership are also made up to the founders, e.g. when the majority of the founders vote for it, because the member is accused of having violated the membership agreements.
- rules on the privacy concerns; since the founders have the right to collect and publish the shared services and resources that are subject of federation, privacy rules for granting or denying authorization to other members can be part of the TLA rules in order to ensure that the federated data is only shared under an award made available from the resource owners, and that it will be refrained from any practices which might actually violate the given agreements.

Members represented as individual companies and organizations may wish to express their own individual policies and cooperations' rules to have control on the access to their federated data. Obviously these rules are made in accordance with the founders' rules and may rely on the following:

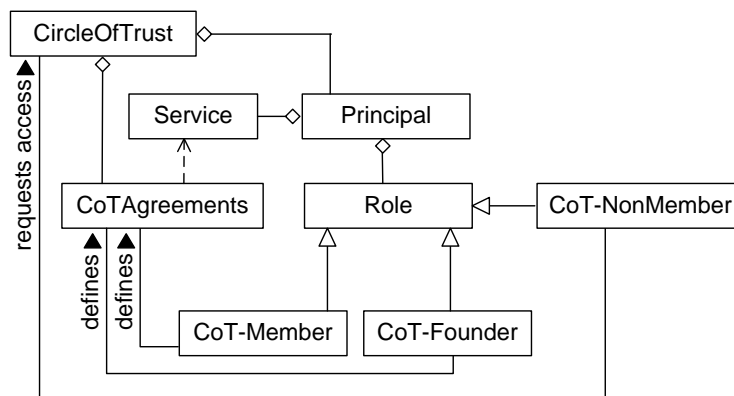


Figure 2.10: Resources being shared among the members in the CoT

- membership rules; including for example the desired duration of the membership or its temporarily suspension in accordance with the CoT terms. Other examples of such rules may cover aspects like retrieving data and services from the federation.
- privacy policies; as it is known in many large networking services, there have been growing concerns about users and providers giving out too much personal information as well as the threat of fraudulent or imprudent data use. As users of these services need to be aware of data theft, their home organizations must establish data protection and data sharing that strike a balance between privacy concerns and the needs of federate personal data in the delivery of public services. However, large services, such as MySpace¹, often work with law enforcement to try to prevent such incidents.

2.1.2.7 Shared services and resources

From the definitions given above, the management of resource sharing in CoT may be distributed among the different kinds of members rather than to be centralized; e.g. as it is known from standard FIM systems, where the users' accounts administration and the resource management are handled by two separate providers.

In the same effort of outsourcing tasks for resources and services management within the CoT, it is, however, substantial to relate them to the members (who either own or simply use them), and define the ownership of the resource especially in cases when there is more than one member which contributes in the management. Note that the ownership of each member's contribution shall be considered in accordance with the formal TLA. In Figure 2.10, we illustrate the ties from the *service* class to the *principal* as well as the *CoT Agreements* classes.

¹www.myspace.com

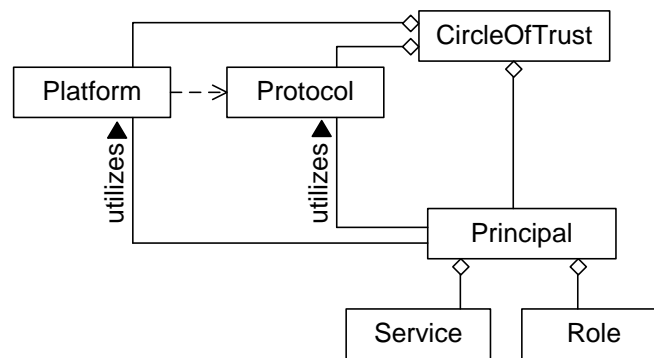


Figure 2.11: Communication trust protocols and platforms of the CoT

2.1.2.8 CoT Platforms

We refer to the *CoT Platform* as the technical shared platform such as a shared web platform that enables principals to authenticate their identities, and accordingly utilize the shared resources. This can be defined as any graphical user interface build upon a set of open protocols (see Subsection 2.1.2.9) that enable any involved member in the CoT, ranging from CoT members to non members, to communicate and share trust data between each other in common shared platforms.

2.1.2.9 Trust communication protocols

The *trust communication protocols* represent a substantial means for establishing trust and protecting information from unauthorized access, use or disclosure. However, they can have different features intended to ensure reliable interchange of data over an imperfect communication channel. Basically, they cover at least two types of trust: (i) trust that designates unknown principals, or groups of principals, so that other trusted principals can establish trust chains to them on the basis of this identification (e.g. public key infrastructures establish trust chains from digital signatures to the signers of the prospective key), and (ii) trust in the way these identified principals will benefit from the assigned rights in the protocol, likewise to Simple Public Key Infrastructure (SPKI) solutions, which identify a set of principals from whom authorization may flow, and interprets each of the certificates as a function of the right the principals can obtain.

Figure 2.11 recapitulates the association between trust communication protocols and domains in conjunction with the CoT.

2.1.3 Classes of CoT

As mentioned in the previous sections, federated environments, defined as a group of organizations that primarily collaborate via dedicated techniques and protocols, is one of the common use cases in IT systems, where trust relationships are mostly needed. Collaborations are usually built through organizations, which come together to cooperate by sharing resources, knowledge and services for enhancing the activities in which

they are engaged. The collaborations in these environments concretize real world scenarios for circles of trust and thus exhibit most of the characteristics of the CoT.

For the duration of these collaborations, trust is especially important as a foundation of any relationship there, because a trust relationship may be used as measure of the trustworthiness of an entity, which can be potentially involved in a cooperation with severe consequences, in case of lack of trust.

Before giving a precise definition of the different classes of the CoT in relation to federated environments in real-world examples, it is necessary to understand their concept and their common properties. Several questions need to be considered, e.g. how can the CoT be built? How are the relationships within characterized? What different kinds of trust relationships are there? How can they be measured, compared and quantified?

This section will highlight three classes of CoT (static, dynamic and virtual). Due to the fact that federated environments may widely include almost any community in the World Wide Web, in this thesis, we primarily consider federated environments that are associated with work-oriented and professional groups such as professional communities and organizations collaborating in well-known resource sharing applications.

2.1.3.1 Static Circle of Trust

Today, technologies are converging to facilitate communication, so collaboration among organizations is seen as a way to glean new insights for reducing costs and raising revenues in several collaboration fields. Structurally, we refer to this type of federated environment, characterized in a static and fixed number of organizations or divisions that primarily articulate the value or the need of trust by setting collaboration standards and agreements, as a *static CoT*.

The empathy that points out the shared reflection of bringing principals (organizations and groups of users) together into static circles of trust via technology and across barriers of organizations is based on several characteristics. In Table 2.1 below, we describe the features of static CoT in light of the definitions and dimensions given in Subsection 2.1.2. An explanation of each feature follows.

Nowadays static CoTs are applied for various purposes. They can be deployed for every aspect of business, academic online learning, healthcare as well as conversations and online conferencing, because trust is important to support the organizations pooled together for the purpose of making large-scale investments. As we can deduce from this table, the trust relationships are quite statically arranged and are mainly based on the security features as well as on a set of comprehensive forms and agreements crafted to meet the needs of the principals in well-known resource sharing situations.

In comparison with the characteristics mentioned above, the static CoT in the LAP project is designated within the *Consortium model*, which is basically appropriate for a small CoT with a static number of participants [VSH05]. That implies, that the CoT-Members are not supposed to be joining and leaving the CoT dynamically over a period of time. This model, as depicted in Figure 2.12 below, provides the members with one consortium agreement to establish the rules, regulations, policies, and guidelines of the CoT. There, every participant may retain control over the CoT to a certain extent.

Static Circle of Trust (CoT)	Characteristics
CoT Principals	The static and known set of organizations collaborating together emphasizes the <i>static</i> feature of the static CoT. That is, all the participating organizations should be known to each other during the creation of the CoT as well as the collaboration process.
CoT Roles	Both roles (CoT-Founder and CoT-Member) are enforced in static CoTs. The founder accommodates usually the role of the administrators or moderator who certify, issue credentials (e.g. define their roles, tasks, and responsibilities) and enrol the groups of users, customers and providers. The diverse group of organizations that are brought together into the CoT for exchanging data, resources and knowledge form the member group of the static CoT.
CoT Trust Relationships	Privacy and Security matters are basically defined in the SLA and implemented centrally in each organization's platform. Additionally, the spirit of trust among the participants can help strengthen the collaboration between them. Often trust is associated with the experience of using the resource, purchasing experience and customer recommendation.
CoT Agreements	Supports collective formal and non formal agreements.
CoT Shared resources	Represents common types of resources being shared among the principals. The combination of collaboration tasks and the number of organizations and users involved in the static CoT is determined by several parameters, such as the nature of the shared resources, communication media adopted, and the capability of the system.
CoT Platforms	Represents the online interfaces used by the principal for data federation and resource sharing.

Table 2.1: Characteristics of static CoTs

2.1.3.2 Dynamic Circle of Trust

The main vision highlighted by dynamic CoTs is that prosperity and competitiveness of organisations and companies depend very much on the way they are able to react flexibly and pro-actively on a constantly changing environment.

In this subsection, we introduce dynamic CoTs, which basically endorse the same characteristics presented in Table 2.1 for static CoTs. However, they differ from static CoTs through the fact that they are tightened by the dynamic nature of the environment's membership, with members joining and leaving the CoT over a period of time. This dynamic aspect may concern the life cycle of the shared resources and services as well.

It is however important to mention that although the participants' membership might be quite dynamic in such CoTs the participants retain the needed knowledge about each other as well as knowledge about the newly involved members. This is because usually

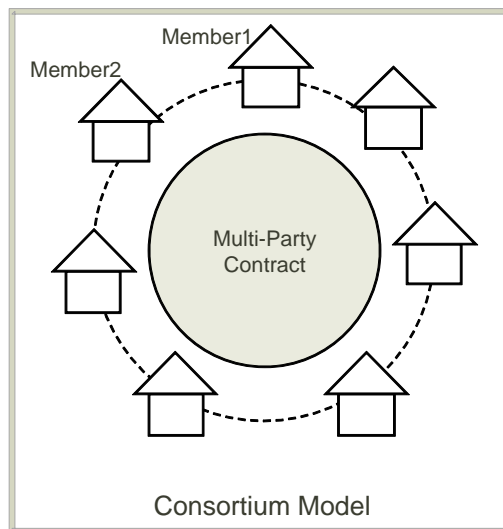


Figure 2.12: Consortium model (Liberty Alliance Project)

the CoT-Founder or the group of founders are responsible for managing the dynamic membership inside the cooperative groups, and correspondingly give the knowledge relationships between the participants.

However, as globalization and run-time communication between organizations in dynamic CoTs move collaboration further into inter-domain directions, dynamic CoTs need to disclose information about the dynamics of the group membership, such as the group size and the number of join and departure members with a special care.

A close definition of dynamic CoTs exists in the LAP project, within the *Collaborative Model*, illustrated in Figure 2.13. With regard to CoT membership, the collaborative model is appropriate for:

- A large CoT in which the parties anticipate that members will be joining and leaving the CoT dynamically.
- A group of CoT-Founders forms an entity that establishes the rules for the operation and governance of the CoT, so that no participant has to bear the burden for the administration of the CoT on its own.
- After the initial phase of role assignment (mainly the identification of the governing entities) and distribution of tasks, the collaborative model provides, accordingly, a single consistent entity with which to contract for the inclusion of new members.

2.1.3.3 Virtual Circle of Trust

Beside the static and dynamic aspects of CoTs, a couple of virtual aspects complement these definitions as well. These virtual aspects are recognized especially in situations when the CoT-Members being indirectly involved in collaboration with merely ambiguous or incomplete information about each other.

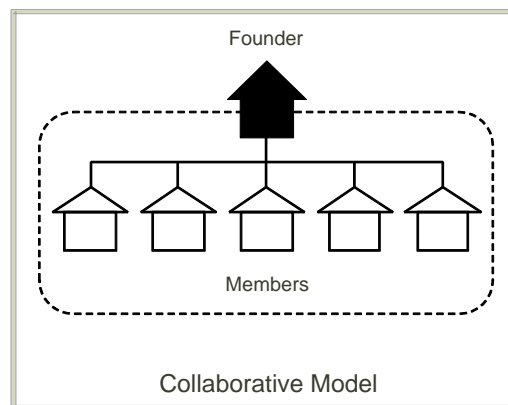


Figure 2.13: Collaborative CoT (Liberty Alliance Project)

As we will discuss in this subsection, virtual CoTs are able to incorporate features of each of the two classes described above, but there are some critical differences that distinguish them. While, for example, they resemble static CoTs in their emphasis on collective agreements and interfaces for participatory communication and resource sharing, they are rather decentralized and the participants may be involved in cooperations and usually not even be known to each other.

While in a small-size CoT (with a limited number of members) organizations may have a good chance to get to know each other, and can therefore have a subjective judgement of the trust level of others. In large-size CoTs, trust analysis of other organizations is a main obstacle for the CoT creation and for building an objective (fact-based) approach for establishing trust relationships among the members.

These challenges are very much reflected in the concept of Virtual Organization (VO) in Grid Computing, where the participants in the virtual organizations can organize themselves dynamically e.g. in a group in order to provide the appropriate service and functionality required at a certain point of time (this applies particularly when the requested service can not be provided by the organization that is in contact with the requester), as it is the case in the On-Demand and Interoperable Grids Model [Pap08]. There, according to the specified requirements and policies, any new VO can be made available and offer its functionalities to every other participant in the environment, which may consist of millions of interconnected participants located behind the Grid nodes.

Although the virtual organization does not have a universal definition; there are many ways in which definitions for virtual organizations and communities can be formulated and automatically derived in Grid environments. An illustrative example is that of virtual organizations for data sharing, where the Grid can be considered as a large distributed data server. For this objective, the virtual organization can be made up of a pool of servers that basically run the same application for data sharing and storage, e.g. GridFTP [GRI]. By means of the VO concept, this Grid application can take advantage of the storage capacity of high-performance servers, and thus choose dynamically the more appropriate available network bandwidth and even distribute the tasks among other servers if needed.

While most of the key characteristics of the CoT are supported (see Table 2.4 in Subsection 2.2.3 where we present a detailed scenario for virtual CoTs), the virtual aspect,

in this regard, is enforced by the fact that the participants might not be known to each other and not aware of the nature of their collaboration parties.

As we will discuss in the following virtual CoT scenario, these collaborative environments might result in an unreliable environment, where undesired behavior from certain participants can be expected. Therefore, mechanisms for trust management such as dynamic behavior control should exist in accordance with the VO.

2.2 Circles of Trust Scenarios

In the following we review the above mentioned selected aspects and classes of CoT by means of three basic real-world scenarios that allow us to deduce the common requirements for assessing dynamically trust across organizations boundaries.

2.2.1 Scenario 1: CoT in academia field - IntegraTUM Project

To illustrate the importance and the necessity of trust assessment in the different classes of CoT, introduced in Section 2.1.3, we present a simplified view of a real-world eLearning scenario in the MNM-Team's environment as an application scenario of the static CoT, which will be faced with dynamic inclusion of unknown users, and thus with the need of dynamic establishment of trust relationships.

Two of the Munich universities, Ludwig Maximilians University (LMU) and Technische Universität München (TUM), offer several joint study courses, e. g. medicine and bio-informatics; students of these study courses are enrolled in both universities and thus must be able to use both universities' IT services, including the learning management systems (LMS).

Additionally, more than 30 higher education institutions (HEIs) in the German state of Bavaria are carriers of the so-called Virtual University Bavaria² (VHB); the VHB acts as a broker between the students and each HEI's local LMS, which results in a highly distributed federated environment with a focus on eLearning services. In the same time, the VHB is involved in several Learning collaboration projects on the national as well as on the European level, such as in the Erasmus Student Network (ESN) project³.

Given the naturally high fluctuation of students and the regular changes concerning which eLearning courses are offered, new technical measures are required to improve the reactivity of ITSM workflows and thus support the underlying business processes. This scenario represents a FE in general, as discussed in Subsection 2.1.1.

Regarding the SLA (detailed definitions to SLA, respectively TLA, in ITSM are given in Subsection 2.1.2.6) for a typical LMS and the privileges derived thereof, we naturally need to distinguish between users and resources. Resources include the various types of LMS content, e. g. lecture notes, exercises, and presentation slides. To handle the masses of users efficiently, Role Based Access Control roles, such as students, lecturer,

²<http://www.vhb.org/>

³The ESN Project is a non-for-profit international student organization, with the mission to foster student mobility in Higher Education under the principle of an access on shared Learning material. ESN Project has more than 12.000 members from 272 local sections in 33 countries working on a volunteer base in Higher Education Institutions, and is offering services to more than 150.000 students. For more information visit <http://www.eu.daad.de>

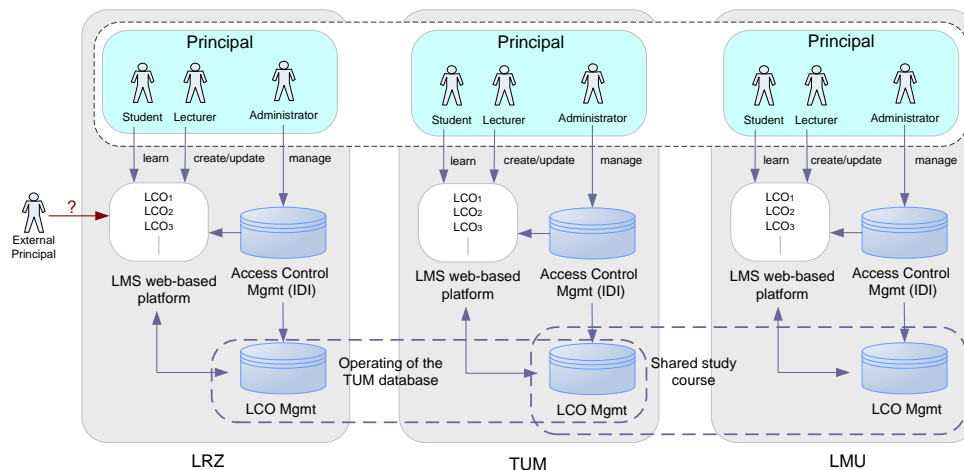


Figure 2.14: A dynamic federated environment for eLearning services

and LMS administrator, are defined. It is noteworthy that the same terminology for at least a subset of the RBAC roles is also used for the description of business roles, which are utilized in the textual formulation of SLAs; for large federations, this implies that a common terminology is required, which is often hard to achieve (for example, the terms student, faculty, staff, and alumni have slightly different semantics in the USA and in Europe).

Accordingly, the main components of the LMS system, presented in Figure 2.14, can be broken down into two categories:

- *Learning Content Objects* (LCOs) basically represent the course material created or coached by the trainers and consumed by the learners. This learning content is usually stored in object-oriented multimedia databases along with various meta-data;
- *Identity Information* (IDI) provides relevant information about the LMS users. Traditionally, the attributes of each user profile object link it to one or more of the defined RBAC roles, which are more efficient to use in access control policies than long lists of usernames that would have the same privileges. The main user's roles in these systems are the (i) *learners* (students, trainees, apprentices, etc), (ii) the *teachers* (who may be the study course authors and therefore owners or simply the teachers of these courses) and (iii) the *administrators* of the LMS platform.

On the basis of this analysis, we conclude that the LMS in this form (i.e. for interactions that *exclusively* occur between the members of the FE) can be viewed as a static CoT because of the static nature of the organizations' membership as well as the management of the internal IDI and LCO groups. Table 2.2 shows some basic mappings between the formal characteristics of the CoT, investigated in Subsection 2.1.2, and those of the LMS.

Circle of Trust (CoT)	LMS system
CoT Principals	Include the Higher Education Institutions that participate in the shared LMS.
CoT Roles	All the typical three roles of the CoT are endorsed in LMS: <ul style="list-style-type: none"> • CoT-Founder is represented in the role of the organization that sets and controls the management rules of the collaborations within the LMS, for example LRZ can be regarded as the founder in some respects. • CoT-Members; all the participating organizations, for example the universities TUM and LMU including their groups of users (teachers and learners) can be regarded as the members of the CoT according to the definitions given before. • CoT-Non-Members: External organizations to which the external users belong characterize the role of CoT-Non-Member.
CoT Trust Relationships	They are principally defined in the SLAs as well as in the static security techniques.
CoT Agreements	Mainly defined in the SLAs.
CoT Shared resources	The learning content objects (LCOs) as well as the IDI who might also be subject of federation.
CoT Platform	The LMS web-based platform mainly used for distance learning.
CoT Protocols	Represented in the security transmission protocols.

Table 2.2: eLearning static online community in light of the formal definition of the CoT

However, as also shown in Figure 2.14, an institution's LMS often is a distributed system itself. In our scenario, the Leibniz Supercomputing Centre (LRZ) operates the multimedia databases and streaming servers of TUM's LMS; these two services are also used by other LRZ customers, which necessitates an additional access control layer on the LRZ side. Furthermore, LCOs are managed by different content suppliers, and trainers and also learners can be affiliated with more than one HEI. In practice, especially concerning the medicine study courses, the LMS must additionally support the handling of third party LCO vendors, external instructors, and guest students.

SLAs exist between TUM and its external suppliers, and contractual frameworks, e. g. for the students, exist; because several study courses cannot be completed anymore without taking tests involving certain eLearning classes, guarantees regarding several classical quality of service parameters, such as service availability and mean time to repair, must be made. The typically short lifetime of eLearning classes, which is about 10–12 weeks, and the skew that all the classes start at the same day at the beginning of each semester, make traditional service level management next to impossible to handle

on a per-service-instance-and-involved-party basis.

However, the LMS systems often do not address only the group of principals identified by their IDI, because external principals may also want to attend a course or look at providing their own LCOs to the LMS system. The group of external principals can be very heterogeneous, as they vary significantly in their prerequisites, their abilities and trustworthiness in their goals of interacting with the LMS system.

Therefore a situation is considered, where a learner from outside the LMS, wishes to use one of the LCO provided by one of the HEI learning platform. Correspondingly, a trust relationship must be established between the external learner and the HEI as a prerequisite for service delivery. It must be ascertained that the learner will provide dependable authentication for accessing the LCO and accurate account data, e.g. for billing purposes. The HEI provider therefore needs to assess the trustworthiness of the requester entity.

Under these circumstances, *trust*, is an important factor in interactive LMS, when the external learners can not be directly identified by IDI, which usually are statically issued from the shared LMS. On the one hand, the LMS provider requires some basis upon which to make trust decisions of the external learner. For example, the provider needs to ensure that the user accessing the system is someone eligible for using the LCO. On the other hand, the learner needs to trust that the provider and the services will protect personal information, and will release information regarding performance for instance, only to those authorized by the learner.

2.2.1.1 Authorization and interaction workflows in traditional LMS

On the basis of similar case studies, which have been investigated in more detail in [BH06b] and [BH06a], a typical LMS request interaction workflow can be described briefly, according to the CoT terms, as follows:

1. The requester browses the service catalogue (CoT shared services) on the LMS platform (CoT platform), which describes what services are available and simple statements on the requirements for receiving the services (CoT context).
2. The requester makes a selection requesting a service, and the service provider (CoT member) is subsequently consulted to deal with the request.
3. Requests with the available credentials (CoT trust credentials) types are collected and checked against the access requirements of the service policy (CoT operational rules and agreements) by means of the deployed mechanisms and communication protocols (CoT protocols).

In the absence of local credentials, usually Public Key Infrastructures (PKI) can also be used to establish domains of trust. For example the *Deutsche Forschungsnetz (DFN)*⁴ is the national German root authority for a PKI hierarchy devoted to HEIs in Germany; so that if the learner comes from a German university he can be trusted merely via this information if the local access policy does not require more access control conditions.

⁴<http://www.dfn.de/>

4. Depending on whether the request and credential types are compatible with the service policy, the requester is faced with two possible outcomes:
 - The service provider grants access when the credentials are identified as IDI, or
 - denies the access, otherwise.
5. In case of granting permissions, the service provider provides the corresponding services to the requester in accordance with access policy statement. Some additional actions, such as logging, altering, may also be executed.

According to these workflows, which basically typify most of the standard LMSs, it is obvious that queries from external learners can either be granted, even knowing that the requester is not identifiable with an LMS specific IDI, and thus risks for altering and misuse of valuable LCOs are probable, or these queries shall be systematically denied with the resulting negative effect on the prospective collaboration. In the next subsection we shall discuss the shortcomings of these authorization models in more detail.

2.2.1.2 Shortcomings and open issues

Centralized access control in FE, as is the case in distributed eLearning systems, presents several shortcomings, particularly when resources and the principals requesting them belong to different security domains controlled by different authorities.

The central role of access control policies in this scenario raises many issues:

1. As we discussed in the previous section, these access control mechanisms make authorization decisions based on the identity of the resource requester. In distributed environments, often the involved organizations use Public Key Infrastructures to build a path of trust. However, when the resource owner and the requester are unknown to one another, for example when the organization to which the learner belongs is not known in the PKI authorities (because each party can decide which PKI authorities it trusts), access control based on identity may be ineffective in this case.
2. There are two more issues next to this aspect, firstly it is not possible to use another type of trusted third party to establish trust between the external learners and the CoT. Secondly, there is no way to specify when to use a zero knowledge approach if no known trusted third party is available at the time of the interaction.
3. In the case the university organization accepts to grant the access for a certain resource or service without a trusted third party or a zero knowledge approach about the external users, there is a lack of mechanisms, by which the users can disclose some of their credentials and possibly some of their privacy policies on these credentials as well (e.g. the user may require a policy that specifies the conditions under which he is willing to entrust his personal data).
4. Obtaining and storing these new credentials represent new challenges on the organization side: How can the users' credentials be collected? How should they

be stored and kept safe from unauthorized use? If there is a need to reevaluate previously provided credentials that are relevant for trust establishment, how can this be done in real time while trust is being estimated?

5. Various access control models that have typically been applied to intraorganizational LMS scenarios have later been extended for interorganizational and federation scenarios for solving such a decentralized trust management problem. Several variants of these standards like RBAC and its successors, e. g. attribute based access control (ABAC), allow the delegation of administration on the one hand and privileges on the other hand; unfortunately, only seemingly they are a good starting point for the inclusion of external entities in FEs, because privileges may only be delegated to those principals which are already known in the federation.

This means that a digital identity that has been created by one of the involved organizations must be assigned to the user a priori, which causes the very same timeliness, cost, and complexity problems we strive to avoid.

By encouraging groups of students, teachers and other individuals to enhance the distance education systems by their participation, these systems are being used to raise the skills and extend knowledge among HEIs, as can be seen in the previous case study.

However, the existing access control solutions therein, firstly, give students and collaborating organizations less opportunity to share users' identities and learning material in automated and controlled manners; and secondly, the administrative overhead for managing the access rights in such a scenario with a multitude of users joining and leaving the CoT is increasing tremendously and is impossible to manage dynamically.

2.2.1.3 Prospective solution

Obviously, a suitable approach to overcome these problems can be achieved by the incorporation of a decentralized management of trust relationships. Instead of or in addition to the static access control configuration, the trust relationships between the users and the organizations inside the CoT can be helpful to identify unknown users, to control their behaviours and to enable them to express and enforce the trust they have in others by means of trust values.

The trust management solution has to investigate many aspects of trust establishment including requirements for trust negotiation algorithms as well as appropriate policy languages for enforcing the safety and consistency of access decisions. By integrating reputation management and distributed audit, the reliability and robustness of these trust negotiations may be enhanced to a greater extent.

Accordingly, this solution needs to cover the following aspects:

- The user should have the possibility to send digital credentials to prove his identity in such a way that these credentials can be verified, i.e. by using a unified format.
- The access control system in the CoT needs to be able to recognize these credentials. For example, in the absence of known credential issuer, additional tech-

niques should investigate whether these credentials were issued by someone who can be trusted.

- In addition to identifying unknown users, the evaluation of their own trustworthiness is just as important. Therefore, this solution should provide a logical framework for reasoning about the trustworthiness of individuals as well as other forms of distributed proof construction, such as criteria for ensuring a continuous assessment of trust.
- If issued credentials are to be shown automatically to other CoT members, privacy concerns may not be neglected. This is because the requester may require a policy that specifies the conditions under which his credentials may be released to third parties.
- The requester and the requested organization for a given service need a communication means that will allow them the opportunity to show each other the credentials that are relevant for the specific request and perhaps also to find out which credentials are relevant for this request.

In the next section we first illustrate common criteria for classifying the requirements for the realization of this trust management solution, and in the following we provide a broad analysis of these requirements into different classes.

2.2.1.4 Classification of the requirements into different categories

On the basis of the given LMS scenario, many requirements arise for the problem of quantifying dynamically the trustworthiness of the external requester (who does not possess static IDI and wishes to use services in the static CoT) and for making service and resource access decisions accordingly.

To illustrate these resulting new requirements and the relationships among them, we recall the traditional centralized access-control architecture within a single organization presented in Chapter 1 in Figure 1.1. We argued that direct trust is not sufficient for the problems stated above, and that aspects from indirect trust (we introduced the different dimensions of indirect trust in Section 2.1.2.1) need to be integrated for extending the centralized architecture for distributed federated environments.

Figure 2.15 gives a brief overview of this extension as well as the classification of the requirements therein. We see that unlike the conventional security management in centralized systems, in which security policies are defined and centrally managed according to a single organization's infrastructure and regulation, the characteristics of Internet-based distributed environment present supplementary challenges. This is because we do not deal with just user authentication and access control to the resources of a single organization. Instead, we deal with a distributed set of interconnected organizations and the sharing of all types of resources that belong to these organizations.

Note that the requirements that can be fulfilled in standard distributed access control systems will not be addressed, and that this classification is given in general, so that it can be applied for the next scenarios.

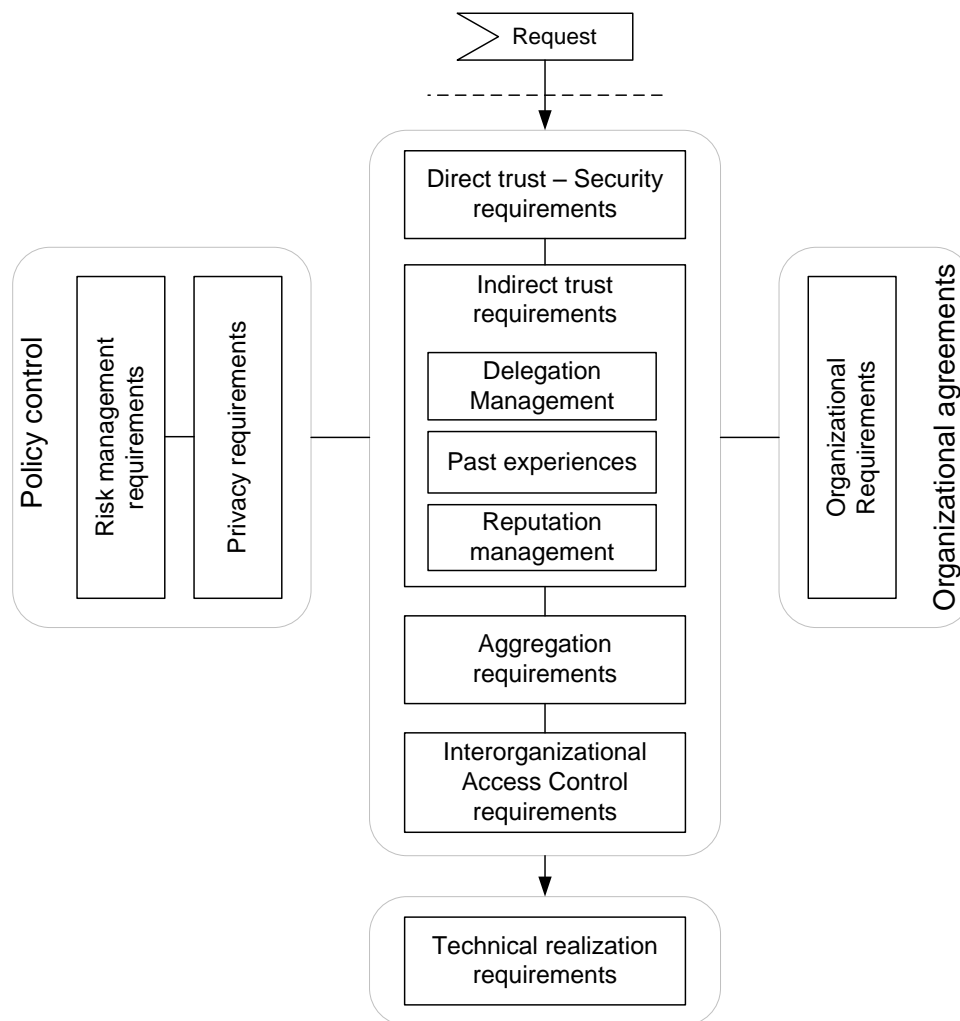


Figure 2.15: Extension of traditional centralized access-control architectures with trust management requirements for CoTs

2.2.1.5 Requirements from the IntegraTUM scenario

In the following, we formulate these trust-relevant requirements into different categories in compliance with the sequence given in Figure 2.15; note that for each requirement we give an abbreviation reference in square brackets in order to facilitate their references in the next scenarios and use cases.

Security requirements in relationship with direct trust

Today, security in relationship with trust is viewed as one of the most important aspects of quality and safety of data transmission and resource sharing.

Subsequently, there are numerous security standards and mechanisms related to the trust issues for intraorganizational solutions, as illustrated in the previous scenario. However, we shall particularly focus on those requirements that extend these *direct trust* solutions and are relevant for trust management in interorganizational surroundings. We delineate

them as follows:

- **Global security policies:** By definition, security policies for an organization –from a high level perspective– address constraints on access control for its users –direct trust– as well as constraints imposed on external users by mechanisms such as locks, keys and fire walls. However in FE such as is the case with the LMS example, often a single organization, for example the HEI TUM is not aware of what security policies should exactly be enforced by an external LCO vendor in relationship with the VHB and across other organizational boundaries in general.

Based on that, policies need to be negotiated and agreed upon by the participating HEIs in order to build up a CoT among them, which should be able to enforce not only individual HEIs' *local* security policies but also those *global* security policies [SEC-Policy].

- **Authentication–Authorization–Accounting (AAA) mechanisms:** In the context of global security policies, mechanisms used for authenticating legitimated users, should also be regarded as such and negotiated in the CoT. Because, for example, the external users requesting access to the LCOs may be identified by different authentication mechanisms (related to each HEI's platform), and the participating HEIs may have different trust views on the used authentication method and protocol, as they may have different ways to trust third parties (for example the certification authorities in the PKI chains). The same requirement applies for authorization and accounting mechanisms [SEC-AAA].

Trust level assessment requirements

From the previous discussions, we brought forward the argument that traditional centralized access-control systems need to be extended with trust management aspects (we illustrated the requirements' grouping for the prospective extension in Figure 2.15).

Trust, in this regard, shall complement direct trust and its security models with the aspects that are related to history from past experiences as well as aspects related to reputations and recommendations from known entities. Moreover, the nature of the CoT as well as the characteristics of the formal agreements among the organizations therein may not be disregarded.

We identify the important requirements for integrating the concepts of trust with respect to authorization models as follows:

- **Trust in intermediaries:** Obviously, communication between collaborating organizations in the CoT may be established through multiple intermediaries rather than directly. In situations where, for instance, the LCO is requested from the TUM portal but is originally provided by another LCO provider, such as a library institution, trusting the requester will then depend on the TUM experience as well as other participants' recommendations for reducing risk and uncertainty.

We conclude that establishing trust necessitates relying on intermediaries' prior experiences and knowledge about the requester; and thus, the trust dependency

and the degree of trust on these intermediaries must be addressed for supporting access control decision making [Trust-Interm].

- **Trust Level:** In the same context, the external principal may be vouched for by one or more known entities (collaboration parties involved in the LMS), which themselves may or may not be members of the CoT; deriving from how trustworthy each warrantor is, a *trust level* for the external principal need to be quantified. This quantification of trust level must reflect various degrees and dimensions of trust, which help ensure that, for example, external entities can access the resource or interfere with the CoT just if a certain trust level is reached.

In the case where all the CoT-Members are ignorant of the external entity (thus have no opinion about the matter of trust in that entity), additional restrictions must be placed for the representation of the trust level in order to take this issue into consideration [Trust-Level].

- **Trust Metric:** The accomplishment of the above mentioned trust level requirement necessitates, in turn, a definition of an expressive metric and data structure in order to encode the trust level according to the requester trust profile, as introduced in Subsection 2.1.2.4.

The way this trust metric can be expressed (either quantitatively or qualitatively), depends strongly on the technical interfaces as well as the communication protocols deployed in the CoT, for example whether it is possible to encode them as numerical values or just define a classification range of trust echelons in specific data schemes [Trust-Metric].

- **Trust Context:** The computed trust level according to the given trust metric may indicate the trustworthiness of the requester for a given situation, for example for a particular service usage or delivery (reflecting for example the reliability of the user when using a specific LCO by the time of the interaction), but principally can not be generalized for any other situation or interaction happening in the CoT. Thus, we argue that the trust management solution has to support this wide range of trust situations that need to be distinctly collected and represented into different *trust contexts* (as defined in Subsection 2.1.2.3) [Trust-Context].

- **Trust Policies:** Subsequently, we conclude that rules and statements to ascertain if unknown entities are to be trusted for a given context and to a certain extent (trust level) are just as essential. These need to be defined in *trust policies*, which specify criteria for trusting unknown entities.

For example, the external users with regard to the LMS may be divided into different groups of students and tutors with different backgrounds and level of expertise. Accordingly, resource owners and LCO providers may wish to express differing beliefs and confidence in these; e.g. conditions for data access only when a given set of attributes is available and a trust level is over a certain limit [Trust-Policy].

Up to now we addressed requirements on assessing trust in general. In the following we shall broaden this study and present requirements for the empirical quantification of trust within its dimension *trust by delegation*. This choice is enforced by the fact that in distributed eLearning systems it is very frequent that teachers might want to delegate

rights, for example, to guest researchers or to assign responsibilities to other assistants in order to perform some tasks.

Delegation and recommendations requirements

In distributed federated environments, usually resource owners (inside the CoT) cannot know in advance which requesters they will have to interact with and which requests they will receive, and subsequently need to use information from third parties who know the requester better. Therefore, there are several requirements on a mechanism that enables a known entity to assert that another entity has the necessary attributes to access the resource.

- **Delegation of credential authorization:** In order to support resource owners in making authorization decisions, mechanisms that rely on delegation of credential authorization as well as third party certifications and recommendations are needed.

In doing so, known users inside the CoT can act as trusted third party entities and will have the possibility to delegate the authority over a credential to the requester entity, as it is the case of the teacher and the guest researcher. Thus, both entities, the requested entity that provides the LCO or the original LCO owner can make access decisions depending on how much trust is put in the third party entity's judgment about the requester [Deleg-Auth].

- **Trust in third party entities:** As was stated in the previous requirement, for handling the above-mentioned granularity and scalability authorization problems, access control decisions need to be based on certified credentials from trusted third parties (TTPs) about the foreign requesters before access can be authorized.

The concept of a TTP has evolved out of the development of public key infrastructures in which cryptographic techniques rely upon the presence of a TTP, which enables the verification of the validity of the keys used to encrypt and decrypt information for and from other parties. However, in the context of resource sharing the requested entity (for example the organization providing the service) may not trust these third parties in any situation, but usually trusts them only for certain things and only to certain degrees (if we recall the example of a teacher and a guest researcher the LCO vendor who is the owner of the LCO may not trust equally all the teachers from the TUM organization for delegating rights on the given LCO). It is, therefore, necessary to investigate mechanisms that support resource owners in choosing the right TTP according to the degree of trust as well as to the giving cooperation situation [Deleg-TTP].

Interorganizational access control requirements

While trust is a statement of belief, access control is a static statement of what the principal is permitted to do, which services can be accessed and under which circumstances. For the case of external principals with respect to the CoT, distributed access control systems, alone, prove to be insufficient for such a task (we will give a broader analysis of distributed and decentralized access control solutions in Chapter 3).

In the following, we shall illustrate the requirements on trust management for extending distributed access control systems to make decisions on the basis of trust analyses rather than on the basis of identity credentials alone.

- **Authority over the resources:** When sharing resources in FEs –with no single central authority– each organization might wish to retain ultimate authority over the resources it allocates prior than allowing users with access to any of the shared resources to have access to all of them, because in doing so this might be inappropriately course-grained access control [Access-Auth].
- **Decentralized authorization policies:** Adopting the trust management approach as an alternative solution for the above-mentioned problems entails that foreign students might be identified by means of intermediaries and may be dynamically assigned with credentials; and this obviously engenders further questions like: Does the set of these credentials and recommendations prove that the request complies with the inter-domain access policy?

Setting up the authorization policies as a *proof-of-compliance* in local domains within each HEI needs to be extended with these cases for the complete LMS in a unified manner. This extension implies identifying the CoT resources –the LCOs as well as the IDIs in our scenario– to protect as well as specify the authorization policies to reason about the degree of protection that each resource needs when exposed to such a foreign request [Access-Policy].

- **Storage of authorization information:** The storage of the authorization information across organization boundaries represents an additional challenge.

Traditionally, this authorization information, e. g, an access-control list, is stored and managed locally by the service in the organization.

However, due to the fact that LMSs evolve rapidly and thus the set of potential actions on the LCOs as well as on the behavior of the learners who may request them are not known in advance, this implies that authorization information (based on the credentials presented by the requester) need to be created, stored, and managed in a dynamic and distributed manner. Moreover, because this information is not always under the control of the service that makes the authorization decision, there is a risk that it could be altered or irrelevant. Thus, mechanisms for managing dynamic storage and update of authorization information must be part of the trust management solution [Access-Stor].

Organizational Requirements

Interorganizational security and authorization guidelines are undoubtedly conclusive for trust management by ensuring the quality of access control on the federated resources. However, as illustrated in Figure 2.15, it is argued, that taking an organizational point of view on trust management yields new requirements with respect to development and establishment of agreements and guidelines among the members of the CoT.

In the following, the primary organizational requirements on trust in CoT are outlined briefly:

- **Trust Level Agreements:** In the previous requirement we argued that in collaborative environment, like distributed eLearning systems, an organization cannot predetermine the users of its resources and their access privileges. Thus, the collaborating organizations need to enforce new agreements among them in

order to take these cases into consideration. The establishment, management and enforcement of these agreements represent a new dimension of collaborative environments.

By definition, service level agreements provide a useful basis for users and providers to assess whether a service delivery capability is likely to be delivered or not. Using trust management approaches as a basis for service selection for newly involved entities in the distributed environment requires the establishment of TLAs as standards (we detailed TLAs in more details in Subsection 2.1.2.6), which may map, for example trust levels for service usage with trust contexts [ORG-TLA].

- **Short setup time:** In this context, the involved entities are bound by contractual frameworks but must support the temporary inclusion of external entities. Therefore the setup of the requested cooperation should be quick. This is because delays caused by the setup of trust and security infrastructure cause opportunity cost [ORG-Time].
- **Simplicity of the guidelines:** It is noteworthy that the guidelines of the new dynamic cooperations must have simple and valid indicators to monitor the actual interaction and to evaluate the resulting outcome [ORG-Simple].
- **Low cost:** Further, the cost of the setup of these kinds of cooperations should reflect the benefit, thus making the former dependent on its duration and business volume [ORG-Cost].
- **CoT integrity:** The integrity of the CoT must not be diminished. A CoT, which by definition has to ensure a secure eLearning collaboration in particular as well as other case-based collaborations between organizations implies common rules and procedures designed to serve as assurance for the members. Therefore, relaxing the standards for the benefit of dynamic cooperation may not defeat the purpose of the CoT [ORG-Integr].
- **No impact on third parties:** An additional aspect deals with the dynamics and processes of third-party interventions and their potential impact on the existing federation agreements. Accordingly, cooperations crossing the borders of the CoT may have an impact neither on participant nor on non-participant CoT relationships [ORG-Impact].

Privacy requirements

In addition to the TLAs, there are, of course, privacy and other concerns for which all of the CoT members must comply with the existing laws and directives. For example, in the former scenario, a student from a German institution might wish to put more privacy restrictions on his federated data, such as he accepts to entrust some of his personal data only if the providing organization belongs to the DFN alliance.

Some of the key requirements for the privacy of federated data include:

- **Data Collection:** Clearly, no entity should hold its own trust value (in which case, every entity would pretend to be the most trustworthy!). Additionally, because of the decentralized nature of the FE, there is no central data repository where it can be stored. Therefore, dedicated privacy policies that cover

issues like (i) what trust information may be collected across domains (ii) where and for how long it might be stored and (iii) how it might be handled, need to be exactly defined and enhanced in the CoT [Priv-Collection].

- **Data Usage:** Particularly, for issue (iii) there is an ultimate need to feature special settings that allow to ensure a certain degree of control over how the trust personal information (this trust information can be represented for example as the trust level given to the learner in a given context) can be shared in the CoT, and under which circumstances it may be passed by. Precisely because the prospective trust management solution might follow information and feedback forwarding mechanisms where the trust information about the entity reaches the requested organization after going through a number of other intermediaries in the CoT. These settings need to be specified within the TLAs [Priv-Use].

Technical realization requirements

This group of requirements addresses essentially issues for the design and the technical realization of the trust management approach that should be based on the previously defined requirement areas. This includes:

- **CoT-Platform integrity:** As we mentioned in Subsection 2.1.2.8, in the CoT, there are numerous software components related to the realization of secure data exchange and collaboration among the members, such as the available transmission protocols for data transfer as well as the facilities for data storage. Taking into account the problems of dynamic and external requests, it is obvious that the existing technical interfaces have to be extended to fulfill the aforementioned requirements. However, since new components have to be integrated within the present platforms as well, the integrity of the whole system may not be affected by this extension [Tech-Integrity].
- **Realization of the trust management solution:** For the realization of the trust management solution, new architectural components need to be explored. They must enable requesters to carry out trust negotiation sessions with the CoT to gain access to resources within a security domain without requiring existing services and protocols to be modified to support trust negotiation natively. These components can be broken down into two classes:
 1. *Trust protocols;* as external principals are, by definition, not known to the CoT members before their first service request, dedicated introductory and negotiation protocols must be adequately deployed [Tech-Protocol].
 2. *Data storage facilities;* the second area of concern for the realization of this architecture involves designing and maintaining data repositories for archiving the trust related information about the entities in relation with the CoT [Tech-Storage].

2.2.2 Scenario 2: Dynamic CoT - Multimedia Digital Library Case Study

In the following subsection we consider trust management issues in dynamic online communities (see Subsection 2.1.3.2). As an application scenario for these communities, we seek to face the problem of trust and access control in digital library domains

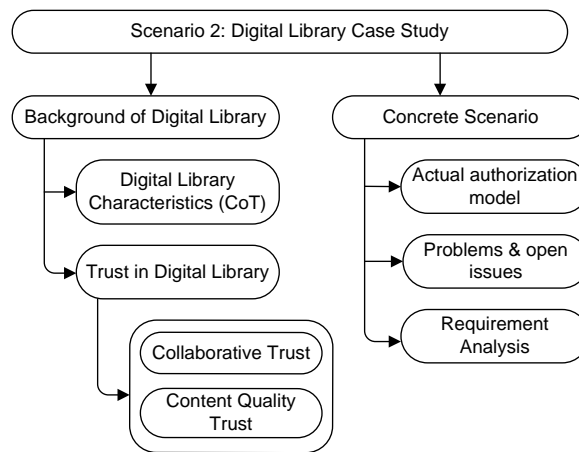


Figure 2.16: Process model of the Digital Library case study

that typify one of the major case studies of a collaborative and evolving information repository that has a variation of quality and coverage of its subject matter.

This choice is based on the fact that digital libraries support a wide variety of applications, ranging from educational and research fields to government and private sector activities. This increased dependence of a variety of applications on digital libraries as well as the distributed nature of privacy and copyright requirements raise the need to develop security models. While there is a considerable amount of research work that focuses on digital library design and efficient data manipulation for providing library services distributed authorization issues and data protection are considered on a small scale.

This section is organized as follows: In the succession shown in Figure 2.16, we first provide information about some of the characteristics of digital libraries and the meaning of trust therein. Subsequently, we present a concrete scenario that investigates trust issues in the web-based content-aware methods for locating, analyzing and presenting the digital data stored across distributed databases.

2.2.2.1 Background of digital libraries

Digital libraries (DLs) are defined as a distribution of autonomous and heterogeneous systems that carry out interactions among information and knowledge organizations, educators as well as end users seeking to access this information. The goal of a digital library is to provide the ability on a global scale to acquire, store, and retrieve information electronically. They deal with large amounts of multimedia information where objects may be stored on a variety of formats and typically come from a variety of sources which may wish to control the DL use (retrieval or modification) or to add value to the content. Users of these systems have a wide variety of diverse backgrounds and interests and usually access the DL from remote sites.

2.2.2.1.1 Digital Library characteristics in comparison with the CoT As stated earlier, the main feature of a collaborative information repository, such

as DLs, is that it may benefit from contributions of a wide diversity of library organizations. These participant organizations, represented by the administrators of the online repositories, generally form the predominant group of principals who create and manage the shared resources in the DL. However, end users, who may or may not belong to these organizations, represent the consumers for whom the information is made available in such an environment.

Circle of Trust (CoT)	Digital Library Environment
CoT Principals	Includes the library organizations (CoT-Members) as well as the end users, who may come from organizations located outside the CoT (CoT-Non-Member).
CoT Roles	The typical three roles of CoT are endorsed in DL environment (CoT-Founders: administrators or moderators of the DL, CoT-Members: Participating library organizations and CoT-Non-Members: Organizations outside the CoT).
CoT Trust Relationships	They are principally based on the privacy as well as the authorization policies that are enforced by the DL system.
CoT Agreements	Represented in the Service Level agreements, the privacy policies as well as the copyright general terms and conditions.
CoT Shared resources	Online articles, documents and other digital material are considered to be the resources being shared among the principals in the DL.
CoT Platform	Represented in the DL frontend as an online platform supplied to the end users for accessing the content of the online articles.
CoT Protocols	Federation protocols as well as any other technical transmission connectors that affiliate the archive repositories and databases can be seen as the CoT protocols.

Table 2.3: Digital Library characteristics in light of the formal definition of the CoT

In addition to the principals, the digital library environment, as we can deduce from Table 2.3, matches several characteristics of the CoT and, thus, can be viewed as an instance of it under some dynamic aspects. In contrast to static definition of CoT, the dynamic facet of it is typified by the fact that not only external users are expected to enter the CoT, but also supplier organizations may join the (CoT-Member) (see Subsection 2.1.3.2 for a detailed comparison).

2.2.2.1.2 Trust in Digital Library Based on that, for trust management in dynamic collaborative repositories, we differentiate between two relevant application areas: *Collaboration trust* and *Content quality trust*. Subsequently, in Subsection 2.2.2.2, we illustrate a concrete scenario that exemplifies these aspects and point out the substantial requirements on a trust management solution for DL environments.

Collaboration Trust

In collaboration trust, questions about how to estimate the trustworthiness and the origin of the users, which may use resources in the collaborative environment, are addressed. That is, due to the fact that online registration is the main requirement for the identification process of library users, every Internet user may create an account therein and get privileges by virtue of the trust of the personal data and the information provided by the user. However, this information may change over time and, therefore, information that was previously contributed by a known and trustworthy user may be outdated or updated by an unknown user.

Additionally, in collaborative resource sharing, such as digital libraries, the collaborative repositories usually contain contributions from other library organizations, many of which will be unknown to other participants as well as to the potential end user. As a result, although DLs have grown in size, their reliability and prevention from malicious and untrustworthy entities may become more and more questionable.

Content Quality Trust

Adjacent to the collaborative trust, *Content Quality Trust* is just as decisive for trust management in online collaborative environments in general and in DL in particular. That is, as collaborative digital repositories grow in popularity and use, issues concerning the quality and trustworthiness of the content information of the document entries grow simultaneously.

In this context, a serious growing problem for digital library repositories has been the issue of estimating the quality of the documents deposited in them. Draft versions, working papers, different formats, supporting material and so on are usually accepted by repositories, but their version status is often poorly described and items are often not linked together appropriately, which, basically, may have a considerable impact on trusting the content of these repositories.

2.2.2.2 Concrete scenario

We consider the scenario of the University Library (UL), which serves as the academic information centre of the Technische Universität München (TUM) and safeguards the scientific literature supply for learning, teaching and research. Furthermore, the TUM University Library provides literature services for the economic and industrial sectors of the public market economy in the free state of Bavaria as well.

One of the major objectives of the TUM University Library (TUM UL) is to strengthen partnerships with other research libraries and to enhance international co-operation with university libraries all over the world. However, due to the growing competition of resource sharing with sinking budgets and information protection, new research challenges in the field of co-operation among libraries and other participants in the information community are growing.

As a member of the Bavarian Library Network and being involved in several working groups such as the *Deutsche Gesellschaft für Informationswissenschaft und Informa-*

tionspraxis (DGI)⁵ or the International Association of Technological University Libraries (IATUL)⁶, the TUM University Library cooperates with partner libraries on both national as well as international level. Particularly on the international level, the TUM UL is an active member of the association *Subito*⁷, which is running a document delivery service conducted by research libraries in Germany, Austria and Switzerland. Subito, as a service provider to research libraries, offers a quick and easy-to-use service delivering copies of articles and supports the lending of books for users located in distinct organizations and countries.

In the following we analyze this scenario with TUM UL as a member of the DL that is promoted by Subito association. We discuss the major problems facing such a collaborative environment composed of a large number of library suppliers, publishers and most notably heterogeneous groups of users. For that reason, this collaborative environment, on the one hand, strives to be secure against malicious use and data corruption, and on the other hand, aims at offering an open access to electronic materials so that information producers can add or update the information any time.

In Figure 2.17 we provide a simplified abstract model that strives to illustrate the Subito open access model across the stocks of the archives of the participating libraries. This model characterizes the interdependencies between the organization libraries (CoT-Principals) who provide the DL documents (CoT-Resources) for end users employing a common web-based interface as a sharing platform (CoT-Platform).

Regardless of where the users come from, any user requiring literature for the purposes of study, for research or lecturing and teaching, etc. can contact Subito directly or access the content via a member library in the DL.

Intuitively, the Subito DL model consists of a set of DL documents, referred to as digital library objects (DLOs) ($DLO_1, DLO_2, \dots, DLO_n$). Typically, DLOs contain information of different media types (e.g. articles, images, videos) which are usually created and owned by single Organizations ($Org_1, Org_2, \dots, Org_n$) (we assume that each DLO is associated with a unique identifier, which is assigned by the system upon the object creation).

Access control is usually performed against a set of authorizations stated by the administrators located in the distinct organizations according to some policies. The authorization, in general, is specified on the basis of three parameters $\langle s, o, p \rangle$. This triple specifies that the subject s , which acts on behalf of the user U , is given privilege p (the access permitted) on the shared object o (the DLO in this case).

However, in practice, each (DLO_i) consists of a set of segments ($S_{1(1..m)}, S_{2(1..p)}, \dots, S_{n(1..k)}$), which are referred to as pieces of information contained in the DLO and which can be identified by names or features. Often different segments of the same DLO have varying protection requirements. Consider, for example, the case of an article from a journal, where its abstract could be made available to everyone, whereas the rest of the article should be made available only by subscription to the journal.

Further, the segment in one DLO could also link to other DLOs as citations. The semantics of a link connecting two DLOs is that the contents of the objects are related.

⁵<http://www.dgd.de>

⁶<http://www.iatul.org>

⁷<http://www.subito-doc.de>

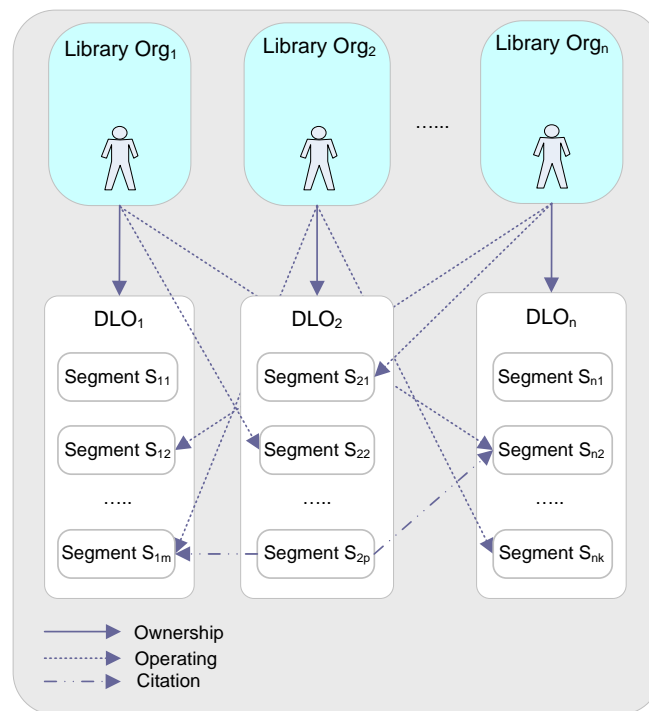


Figure 2.17: Presentation of an exemplary Digital Library open access model

For example, a DLO representing a scientific article may contain links to publications on the same topic. On the basis of these relationships, we deduce three types of links between DLOs:

- **Ownership links** associate organizations to the DLOs they create and consequently own. These links are represented as continuous arrows from Org to DLO . Note that the semantic of these links is directly implied with full rights on processing any operation on the DLO including delete rights.
- **Operating links** associate users from different library organization to DLOs or segments of DLOs, and whose organization does not necessarily own the prospective DLO. The rights coupled to these sorts of associations include mainly reading as well as changing the content of the DLOs according to the roles assigned to the group of users as well as the privacy policies such as the copyright restrictions. These links are represented as dotted arrows from Org_j to S_{n_k} .
- **Citation links** as we mentioned earlier, these links associate segments of DLOs to other DLOs through content links and citations.

2.2.2.3 Actual Subito DL authorization model

The authorization model of Subito DL is mainly based on the categorization of the users into different groups. That is, in order to find out exactly which user may access which DLO depends on many criteria. It depends, for example, on whether the user is resident in one of the German-speaking countries or lives abroad.

However, the main differentiation that requires a trust management solution is made between (i) direct users, who belong to known institutions and subsequently can prove their identities through the federation connectors and (ii) library users who simply come from everywhere in the world with an electronic client profile as a unique proof or assurance of the correctness of their profile data.

The group of external users actually form the majority of the clients of the Subito DL. This is because the decisive political goal of Subito DL is to assist non-commercial users as well as private individuals first and foremost by offering them an affordable professional document delivery service. In addition to these groups of users, it is, however, also possible for external libraries to register and thus become a library supplier member in the collaborative document sharing and information service delivery.

In the following, we briefly present the two registration processes for both users' registration as well as library membership registration:

2.2.2.3.1 Registration as a client As we mentioned earlier, Subito Library Service is exclusively reserved for library users, but due to the different regulations on pricing and royalties, this group of users can also be divided into subcategories:

- **Non-commercial Category;** users falling into this category are pupils, trainees, students, university and college staff as well as employees of research institutes.
- **Commercial category;** employees of commercial or industrial institutions, corporate libraries, self-employed people, and other commercial customers fall into this category.
- **Private Category;** all private individuals without a known institution affiliation are part of the private category.

In this regard, online registration by the user is a requirement for using Subito DL Service. When registering, the user will be assigned to the relevant user group according to the information he provides in the client's profile form. In this information, he has to precise the country where his residence or the institution is located as well as the respective relation to the institution. In case the institution is not available in the list of known institutions, the user has the possibility to enter a new institution by giving a short description.

During the registration process, the user is, additionally, required to confirm personally that he will comply with the copyright laws and agree to the general terms of business. That is, once registration is complete, the user will be assigned with appropriate rights for using the Subito services such as ordering articles and lending books.

2.2.2.3.2 Registration as a library institution In addition to the user's registration, external library institutions may join the supplier group in the Subito DL Service as well. Under some particular conditions, the administrators may register their libraries when it can be ascertained that the institutions are largely financed through public funds and are affiliated with the national or international lending service.

Moreover, this manual registration process does not only apply to state libraries or university libraries, it applies to regional libraries and special libraries as well. Excluded

in any case should be corporate libraries and libraries belonging to commercial or for-profit purposes.

On the technical level, for both types of registrations, such a DL model is usually implemented as software engine that runs on one or more web servers. In the following, we briefly quote some of the basic features for the Content Management Systems (CMS) of the realization of the shared DL storage system for both user's data as well as the digital material which is subject of consumption:

1. The content is initially stored in a shared file system among the library institutions, and potential changes to the content are stored either locally by the library supplier data stores or directly from the Subito DL system.
2. Usually, the DLOs are embedded in hyperlinks so that citation analysis and interrelations between the documents can be conducted.
3. On the basis of the assigned roles and rights, there are individual users who might have permissions for entering new content in the DL system as well as modifying some DLOs
4. Authorization aspects, in relationships between the prospective users and the DLOs, are actually tuned on the basis of the restrictions and regulations (often defined in the copyrights agreements), which are imposed, in hierarchical sequence, generally by the Subito DL system as well as locally by the suppliers and the publisher organizations. Usually, the majority of these principal groups will have expertise levels that are untested and unknown reputations to the end users. Thus, trust management has become a critical component for designing such an open editing platform.

Figure 2.18 depicts an exemplary authorization hierarchy concept for Subito DL. As illustrated, authorization concepts towards the top of the hierarchy are more general and common to most DLOs, whereas the authorization concepts toward the bottom of the hierarchy are more specific to the organization members of this collaborative environment. For instance, a restriction is imposed on the delivery of graphic files (PDF files), which will then only be permissible if the direct publisher does not offer access to the same article online.

As we will discuss in the following section, the above observations and analysis motivate us to revisit the problem of access control in decentralized and open systems as we have it in digital library systems. We believe that a trust management approach is a step in the right direction to complement decentralised access control systems and to recover from the given deficiencies.

2.2.2.4 Problems and open issues

In FEs, in particular in DL and open editing platforms, the costs of long-term information management remain relatively speculative. This is because, on the one hand, the Interorganizational distribution of materials, capabilities and expertise has the benefit of helping to ensure that digital materials survive into the future by maximizing the benefits of cooperation and maintaining a level of redundancy that prevents total loss of the digital shared material due to the breakdown of any one node.

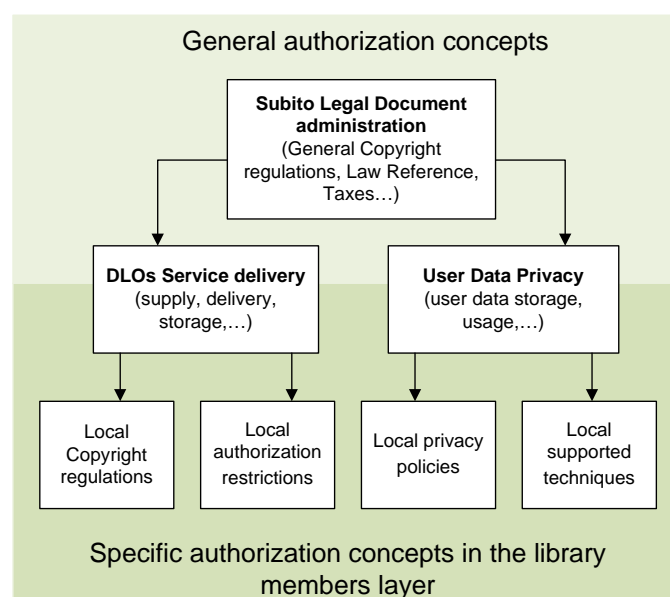


Figure 2.18: An exemplary representation of the conceptual authorization hierarchy in DL

However, on the other hand, for an outsourcing relationship to work successfully, trust has to be established between the entities involved. The trust relationships among the library organization partners, are abstract notions that result from the answers to questions such as how to explore the various mechanisms by which trust can be established to other partners.

A good deal of this discussion revolves around the two aspects of trust, which we already introduced in Subsection 2.2.2.1: (i) Collaboration Trust faces the problem of trusting unknown entities through the ways numerous monitoring, logging and reporting functions report on their own behavior. In our scenario (presented in Subsection 2.2.2.2) this type of trust applies for both external library users as well as external library organizations that join the DL. Additionally, (ii) Content Quality Trust addresses questions of trusting the quality information concerning the DLOs, for example, the quality of an article since it may be published and modified by many sources with varying degrees of reliability (we will revise most known approaches for trust management in DLs in Chapter 3).

We summarize the shortcomings of DL authorization models in reference to these two trust aspects as follows:

Collaboration Trust: Trusting external end users?

- As stated earlier, access permission in these environments is given on the basis of the information entered by the user, because registration by the client is a requirement for using the Subito DL system. By registering, the client consents to electronic storage of details in the client profile, and assures not to provide any incorrect details for the client profile. Although these authorization models accelerate the business process in open systems to a great extent, it has a number of shortcomings:

- The user’s data, strictly speaking, does not bind a user to its purported behavior or actions, and it does not guarantee that its bearer really satisfies the claims in its profile. For example, it is not possible to verify whether the user really belongs to the institution cited in the profile, and whether the given institution effectively exists.
 - The details relating to the user’s delivery and invoicing address need to be filled out with care and updated whenever changes occur as these details help to avoid both documents and invoices being sent to the wrong address. In case the client triggers deliveries by providing wrong information in the client profile, Subito may block the client and exclude him from further deliveries. However, this happens usually after several warnings and inquiries, which might take time in which the user may continue accessing the DL resources.
 - Additionally, the provided profile does not convey any information about the behavior of the bearer between the time the information was entered and its use.
- Another issue is that these access control models do not keep track of the user’s behavior history. Access permission is given on the basis of the information presented in the user’s profile. Either the user’s information is accepted and required privileges are subsequently allowed, or the information is rejected and the user does not get the access rights. Thus, good behavior by the user cannot be rewarded with enhanced privileges nor bad behavior be punished.
 - Often in DL systems, a digital rights management system (DRM system) shall be used for delivery of DLO electronically, e.g. by e-mail or per FTP. However, the DRM system in collaboration with the partner library organizations shall put restrictions on the usability of the copy supplied by these means. For example, the document may be printed only once and finally must be deleted immediately after that. Here, it is obvious there is no efficient control mechanism over the way DLO may be used after permission has been granted.
 - In the same context, there are no ways to verify whether the users will keep the ordered copies exclusively for their own use and will not make them available to third parties; as well as forwarding copies only in a manner which the copyright provisions of all partners allow. This problem has a particular significance, when the legal relationships of the parties can not be governed by national laws.

Collaboration Trust: Trusting external libraries?

Several DL systems such as Subito and IATUL represent a voluntary international and non-governmental organization of a group of libraries, who wish to provide services, not only to the teaching and research staff and students of their own university, but also to other universities and research institutions in different countries, thus building a FE.

Often represented by their library directors or managers, which have responsibility over information services and resources management, these DL also welcome library organizations outside the FE to become a member and supply services. In this regard, trusting the new involved organizations is just as important, because a bad reputation of a single organization may considerably influence the trust in the whole DL system. We

now quote some of the deficiencies of DL authorization models that can lead to such a situation:

- In the general terms of use, among the partner organization, it is usually required that the data provided for registration and within the scope of an order shall be stored by Subito (Subito as an organization plays the role of the CoT founder) for order processing including invoicing and delivery aims. However, in case this information has to be passed on to one of the supplier libraries, – for instance when the DLO can not be delivered by the Subito organization directly – the Subito system will have, subsequently, no control over the way this personal data will be treated afterwards. For example, it can not be controlled if this data will not be used for other advertising measures.
- Including new external libraries in the FE implies approving access as well as modification on the shared resources (such as update of existing documents with newer versions, etc). Although SLAs and contractual frameworks exist between Subito and its external suppliers for managing these cooperations, it is not possible to detect wrong changes or updates whenever they occur on the DLOs. As wrong updates may considerably influence trust in the partner organization, we conclude that relying on the traditional service level management solely makes it next to impossible to handle trust management problems in DL systems.
- Similarly, reliability guarantees regarding several classical quality of service parameters, such as service availability and mean time to repair as well as other organizational aspects affect trust as well. However, it is apparent that the principle of the static contractual agreements disregards the necessity of preventing errors and non-reliability of the organization members by means of trust management paradigms.
- An additional challenge that faces the end users in relationship with the newly involved organizations is the lack of intuitive ways for evaluating indices of trust-worthiness regarding the article provenance, sources used and recommendations about content information manipulation.

Content Quality Trust: Trusting the quality of the DLOs?

Content quality trust is of special relevance for higher educational programs, because the more quality content is provided in the DL, the more the teachers, students, and other end users trust the system, consult the course materials digitally and recommend the articles on a continual basis.

However, there is a greater challenge for finding an acceptable definition for the Quality Content in distributed storage repositories, as it may have different meanings in different use cases:

- With the aim to establish trust concerning the content quality, it is crucial that each original supplier organization maps the documents accurately to a unified quality content parameter scheme (for example with defined dates, identifiers, version numbering, version labels or taxonomies, etc). Due to the large number of data stored in digital libraries with heterogeneous data models, this issue needs to be considered with care.

- The *time factor* is one of the most important content quality parameters, because in order to include the notion of trust and reputation management on DLO layer, a time stamp for making accurate trust decisions by keeping changed reputations and feedback up-to-date may be helpful. Obviously, this hypothesis collides with DL concepts, where DLOs change rapidly and thus recommendations could considerably lose their effectiveness, because the growing numbers of articles and their increasing fragmentation require an increasing number of ratings to keep recommendations significant.
- *Competence aspects*; openness is a feature that is being adopted as the basis of how various groups of authors and end users operate on shared documents in several open digital libraries. Obviously, *closed* collaborative environments are more secure and reliable but grow slowly, whilst more open ones grow at a steady rate but generally result in being an easy target for vandalism.

A clear example of this comparison would be that of the *World Digital Library*⁸ and the *Oxford Text Archive (OTA)* project.⁹ The first is rather open and typified by an open access to its content repositories to the general public, and with few restrictions for external institutions in becoming partners of the World Digital Library. This aspect makes it grow rapidly, whilst the latter is accepting deposits into its collection under a peer review for each individual deposit. The review process requires a detailed biography of the authors to prove their competencies and requires that the deposits are of sufficient quality and come with good documentation. Consequently, this may affect the growth of the DL but creates an almost *vandalism-free* collaborative environment.

2.2.2.5 Prospective solution

Based on the problems discussion hitherto, we conclude that collaborative and content sharing in DL make new demands for managing trust-related principals' behavior as well as content quality. Although actual DL CMSs deal, to some extent, with trust issues such as blocking suspicious users, there is still a growing need for effective strategies for managing the issues stated above.

In this thesis, we will demonstrate that the solution for the discussed problems can be achieved through an enhancement of the trust paradigm in these models with much richer multi-dimensional trust aspects.

Even though the proposed solution may benefit from existing authorization models, it has a number of relevant differences. First of all, it must support the notion of several credentials and past evidence as the basis for identifying external entities; by contrast to other models that use user-ids or groups as authorization subjects. Another relevant difference is that our solution must be able to work in open and distributed environments, and not limited to closed and controlled environments.

In the following we provide some aspects of our solution and show how it may cover the deficiencies mentioned above:

Collaboration Trust: For this type of trust we shall extend the solution we proposed in the IntegraTUM scenario in Subsection 2.2.1. There, we enumerated some relevant

⁸<http://www.worlddigitallibrary.org>

⁹<http://www.ota.ahds.ac.uk>

requirements on how the solution may extend traditional centralized access-control architectures with trust management aspects (all details are given in Subsection 2.2.1.4). However, for the current DL scenario, this solution needs to consider these additional issues:

- In the IntegraTUM scenario we already brought forward the necessity of integrating trust aspects within the authorization models in use. We showed a set of requirements on quantifying and encoding trust into different levels by means of trust metrics. Further, we also reviewed requirements on delegation mechanisms that enable known entities to recommend the inclusion of unknown entities.

However, the concept of DLs unequivocally requires further dimensions of trust. Subsequently, we need to extend our solution with (i) mechanisms for estimating trust from past evidences that rely on the hypothesis that trust can be seen as successor predictor of the entity's future behavior based on past evidences, basically extracted from past interactions with that entity; and (ii) reputation management mechanisms that may significantly increase the trustworthiness of the participants in the DL environment and, thus, ensure the quality of cooperation therein.

- This solution needs to support aggregation mechanisms for collecting, interpreting and aggregating the trust values resulting from distinct evaluation techniques. This includes in particular an investigation of appropriate schemes and unified scales that take care of all these aspects.
- Since we investigate access control decisions for users that have not been authenticated locally, it is difficult to define a policy based on authentication at design time since the potential users are unknown and continuously changing. Thus, our solution needs to explore how access policies can be defined with regards to the computed trust levels, the managed resources and the security requirements related to these resources.

Content Quality Trust:

Besides, the solution for the above named problems should also consider the Content Quality Trust. The intent, in this regard, is to accomplish a trust-related resource description that promotes better practice for shared repository management.

- It is, therefore, necessary to provide resource owners in the DL with unified ways for describing their resources. This description should include a synopsis of relevant quality parameters that enable an evaluation of the federated resources' quality, and a clearer understanding, for example, of version relationships as well as better version identification of digital objects.
- Automatic tools that check the compliance of the resources with the given quality parameters are relevant in the context of Content Quality Trust. Additionally, exchanging feedbacks about the resources may be helpful as well.

2.2.2.6 Requirements from the DL scenario

We now explore the requirements on the solution presented in the previous section. For the classification of the requirements on Collaborative Trust for both external users

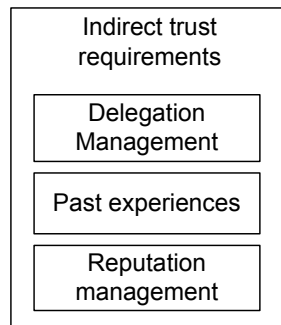


Figure 2.19: Requirements for indirect trust

and libraries, we shall follow a similar methodology as the one presented in Subsection 2.2.1.4. Note that we will mainly focus on those requirements that were not discussed earlier. Moreover, a new requirements' set will be dedicated for Content Quality Trust.

2.2.2.6.1 Requirements for Collaboration Trust In the IntegraTUM scenario (in Subsection 2.2.1.5), we discussed requirements on assessing direct and indirect trust. There, we presented requirement on quantifying, representing and storing trust values in general as well as some extra requirements for assessing trust within one of its dimensions, namely trust by delegation.

Beside these requirements, in the context of the DL scenario we shall broaden this study and present requirements for the empirical quantification of trust within its two other dimensions, past experience and reputation management. Figure 2.19 brings back the representation of indirect trust within its three distinct dimensions.

Past experience and auditing requirements

For protecting resources from unauthorized usage, intraorganizational access control mechanisms aim to prevent illegal actions a-priori occurrence, i.e. before granting a request for a resource. In our scenario, however, the access decision can not be made on-the-fly. This is because, on the one hand, the policies may be created in multiple domains and they are therefore often unknown and incoherent. On the other hand, they may be created centrally by the founder of the CoT (Subito organization is the founder in this regard) according to some global policies that are stipulated in the CoT

For these reasons, it is obviously necessary to control the compliance to the access policies by means of a formal *audit* procedure, by which users may be audited and asked to justify that an action was in compliance with a policy. That is, if a similar access request from the same user reoccurs, a-posteriori access control can be helpful to tell about the trustworthiness of the requester from the audit information. The main requirements for this trust establishment process are summarized as follows:

- **Audit Information:** Audit systems usually observe critical actions within a single organization. In the DL scenario, there must be a sufficiently comprehensive audit trail, which makes this audit information available to the libraries that take part in the DLO sharing collaborations. The audit information should

contain the relevant details about the actions and the identity of both the users as well as candidate libraries executing them.

In addition, there must be also some mechanisms to verify whether this information is not tampered or bypassed [Audit-Info].

- **Audit Data Evaluation:** Basically, the aim of auditing and reporting of past experiences is to estimate trust of unknown entities. Therefore, procedures and mapping techniques that induce and represent the degree of trust in these entities (e.g. as numerical trust values) from the audit information are required [Audit-Eval].
- **Audit Data Storage:** An additional requirement deals with storing this audit data. This is because many questions relate to whether this information may or may not be accessible by other CoT members. There is, subsequently, a need for a distributed data store in that the CoT member's audit information can be made available to others and, at the same time, must be protected with adequate access rights [Audit-Stor].

Reputation management requirements

In Subsection 2.1.2.1 we stated that trust is conceptualized as a multidimensional construct consisting of dimensions such as trust by delegation and trust from past experiences. Trust by reputation is an additional emerging dimension of trust, especially in the commercial and business environments (such as in eBay and Amazon market places), where successful commerce relies heavily upon the reputations that the different parties acquire through their dealings with each other.

In analogy with the DL scenario, reputations, basically represented as ratings given by humans, may be of great help for the above discussed problems in order to reflect the reliability of the users as well as the quality of the shared content. Accordingly, for establishing trust by reputation, several requirements have to be considered:

- **Reputation Value:** Reputation of an entity is usually inferred from the opinions and ratings from sufficiently trusted entities as well as conclusions drawn from observations of previous interactions in which this entity was involved in.

In our DL scenario, the reputation of a DLO might be, for example, based on the number of users having downloaded and used the DLO. We argue that, given the diverse skills that such collaborations involve, reputations may be significantly helpful in deciding which partners to cooperate with.

In the context of a circle of trust, these reputation data need to be summed up, analyzed and aggregated into feedback rating values, which in turn, indicate the trustworthiness of the reputation value's holder [REP-Value].

- **Reputation Metric:** In analogy with the requirement [Trust-Metric] from Subsection 2.2.1.5, a metric for encoding the reputation values is just as important, because it allows participants to rate each other by submitting a comment and a rating according to a standardized scale. As we will discuss in Chapter 3, there are several studies around designing reputation metrics. The choice of an appropriate metric, however, depends on the existing CoT resources and infrastructure [REP-Metric].

- **Reputation Context** Depending on the scenario of the collaboration, the reputation values may have different aspects, which comprise different indicators (for example referring to a certain action, or a set of actions, on the DLO). In order to strengthen stakeholder trust, it is of critical importance to find out which indicators of each reputation aspect exert what kind of influence on stakeholder trust. We refer to these indicators as reputation context, and we argue that with this knowledge, CoT can create a more purposeful and effective trust management paradigm [REP-Context].
- **Credibility of the ratings**: The credibility of the rating is a substantial aspect for building trust by reputation, but is very hard to detect. We conclude that mechanisms for checking the credibility of the rating as well as the context of the transaction are requireable [REP-Cred].
- **Recentness**: Providing entities with the most recent feedback given to the partners they are currently interacting with is significantly helpful to increase trust. However, mapping the time stamps to the ratings that should be classified as a distribution of all previous ratings over the context of the interaction turns out to be more efficient [REP-Recent].

Trust value aggregation requirements

We presented an extended trust management view based on two dimensions for collecting and accessing trust with regard to the CoT. However, it is obvious that the trust values resulting from the related schemes and metrics need to be aggregated to a certain extent.

Under these circumstances, a simple presentation of all trust values, carried out over all recommendations and participants' reputations, is an inefficient method for a dynamic access-control-decision-making. In the following, we present two requirements for aggregating the trust data:

- **Trust values collection**: In order to aggregate the different trust values, first and foremost, it is necessary to collect this data and to investigate the way it should be represented to the CoT members (for example in a sort of descriptive list), so that this representation matches the trust dimension and invokes each trust context contained therein [Aggre-Collect].
- **Aggregation scheme**: Moreover, aggregating the results provided by the varied trust assessment mechanisms invokes a conception of a distributed schema associated with the data storage software in use in the CoT. This conception also involves checks about the conformity of the data types as well as checks for dynamics updates when the schema need to be extended with new trust attributes [Aggre-Scheme].

Organizational Requirements

Most of the organizational requirements, we presented in scenario 1 in Subsection 2.2.1, such as [ORG-TLA], [ORG-Time], [ORG-Cost], [ORG-Integrity], and [ORG-Impact] apply to the DL scenario, because similar to the previous scenario, the DL scenario

consists of a network of independent and even geographically dispersed library organizations that collaborate with each other by sharing business processes and DLO resources.

Therefore, the success of this type of FE, which has similar roles and requirements with regards to the CoT, does not only rely on secure resource sharing but it depends on the security and organizational agreements to a major extent.

Privacy requirements

In this scenario, we provided the evidence that the different dimensions of trust, such as past experience and reputation management, can be applied for quantifying the trustworthiness of an entity.

However, performing the corresponding mechanisms of these dimensions, such as opting for managing users and library organizations reputations, implies that the reputation information has to go through a number of intermediaries (ranging from DLO vendors to end consumers) in the CoT. Thus, similar requirements on privacy ([Priv-Collection] and [Priv-Use]) pertain to this scenario as well.

Interorganizational access control requirements

Most of the requirements from the first scenario, with respect to Interorganizational access control requirements ([Access-authority], [Access-Policy] and [Access-Storage]), apply for the DL scenario. However, a notable difference is typified by the segmentation aspect of the DLOs. This aspect leads to additional requirements:

- **Segmentation aspects:** In the DL scenario, the authorization model may use fine-grained access control for protecting DLOs as well as their segments with, possibly, different access permissions. As we showed in Subsection 2.2.2.2, the differentiation between the links related to the DLOs/Segments of DLOs actually evolves from the corresponding variable permission rights.

This feature for deployment in a highly dynamic collaborative environment poses new challenges. It is, therefore, necessary to investigate decentralized access management procedures that take into consideration that the protected resources may have a set of permissions associated with several segments, and thus, must keep the same protection rules when they are passed on to other CoT members in the FE [Access-Seg].

- **Credentials:** In analogy with the distributed eLearning scenario, actual access control models for DL environments need to be extended with mechanisms that enable requesters to provide credentials, for example from trusted third parties to prove their identities, and accordingly gain access which is based upon it [Access-Cred].

Content Quality Trust requirements

- **Content Quality Parameters:** Users in different roles (end users, designers and administrators from collaborative organizations) are the main actors that exploit the DL functionality for providing, consuming and managing the DL content. Trusting the behavior of these users as well as the content of their resources, strongly depend on their input and the changes activated by them in relationship with the content quality of the DLOs.

Trusting the quality content comprises checks on the content quality parameters of the DLOs. These parameters could be, for example, the result set format, the language, the document model, reference, version number, etc. Accordingly, content quality parameters that enable to build trust from that need to be defined in a standardized scheme [Content-Quality].

- **Reputations through citation:** When the DLO, for instance, an article is cited by other authors in other articles as an information reference, this may show its quality. This citation may be seen as a good reputation of the article. In some settings, end users may be more encouraged to rely on the content of an article if it is cited by others. Consequently, this reputation information should be represented and visualized in an intuitive manner [Content-Rep].

Technical realization requirements with regard to Content Quality Trust

- **Space Complexity:** Attributing trust values in a content repository of the size of DL might be a large task. Besides, considerable care has to be taken because the trust on the content objects must be assigned to the owners of these objects [Storage-Complexity].
- **Automatic Monitoring:** The risk that inappropriate or undesired content shall be inserted in open research editing systems like DLs is quite high. As these systems grow very fast, any level of manual monitoring, such as notifying the owner of the documents, may not be sufficient since it will not be able to scale with the content size.

Therefore, automatic methods for data quality checks are required to improve the administrator's abilities to monitor updates and to help manage instantaneous warnings [Storage-Monitoring].

- **Trust Storage Conflict:** The first requirement of storing trust information depends strongly on a second concern, which addresses issues for storing trust relations between authors and the aggregated degrees of trust inferred from the content objects. This additional stored content, however, may lead to conflicting results about the trustworthiness of authors against that of the document content.

In this context, we face an additional problem for representing and checking the correctness of the estimations resulting from the aggregation mechanisms [Storage-Conflict].

2.2.3 Scenario 3: Virtual CoT - DEISA Grid Project

As an instance of a virtual CoT, we choose to study trust management problems in the Grid computing environment, as these environments characterize open distributed systems in which autonomous participants may collaborate with each other using specific mechanisms and protocols.

In general, the participants have different aims and objectives, have different capabilities for offering services and can join and leave the Grid environment any time. Because of this high mobility, often the participants do not have sufficient knowledge about their collaboration partners. As a result, trust management represents a major requirement for enhancing reliable collaboration among partners in Grid environments, without which it is quite difficult to rely on the outcome of the collaboration process.

2.2.3.1 Background of the DEISA Grid project scenario

DEISA (Distributed European Infrastructure for Supercomputing Applications)¹⁰ is a consortium of leading national supercomputing centres for enhancing European capabilities in the area of high performance computing, and this is done by deploying a distributed supercomputing environment with continental scope.

In the DEISA Project, online computation and storage services are offered as services supported by a pool of distributed computing resources and high-end platforms that are tightly coupled by a dedicated network and supported by innovative system and grid software. In the scope of Grid Computing, the LRZ (Leibniz Supercomputing Centre) participates as an active member beside several other project partners such as German universities, European computing centres as well as several international research centres worldwide.

In the form shown in Figure 2.20, Grid computing provides a framework to end users for exploiting the resources offered by different organizations in a completely transparent manner, so that requesting a service at each single organization is not needed anymore. In the following, we provide a simplified workflows view starting from the client request until the execution of the requested task on the real organization's side, which effectively provides the service:

- The client is responsible for initiating the session by choosing the desired task from the *Self services portal* and sends the corresponding job to the Grid Middleware. Among the provided services, the client may also have the possibility to send his single preferences on data privacy concerns, for example on hiding and protecting data from a certain use.
- Scheduling and management tools within the *Grid Middleware* collect the requests and performs the appropriate *Grid application*, for example for a data storage task, the Grid application strives to take advantage of the storage capacity of high-performance servers and available network bandwidth.
- With the growing size of the Grid, the number of applications that profit from this technology, vary from simple data sharing, to expensive simulations. Therefore,

¹⁰<http://www.deisa.org>

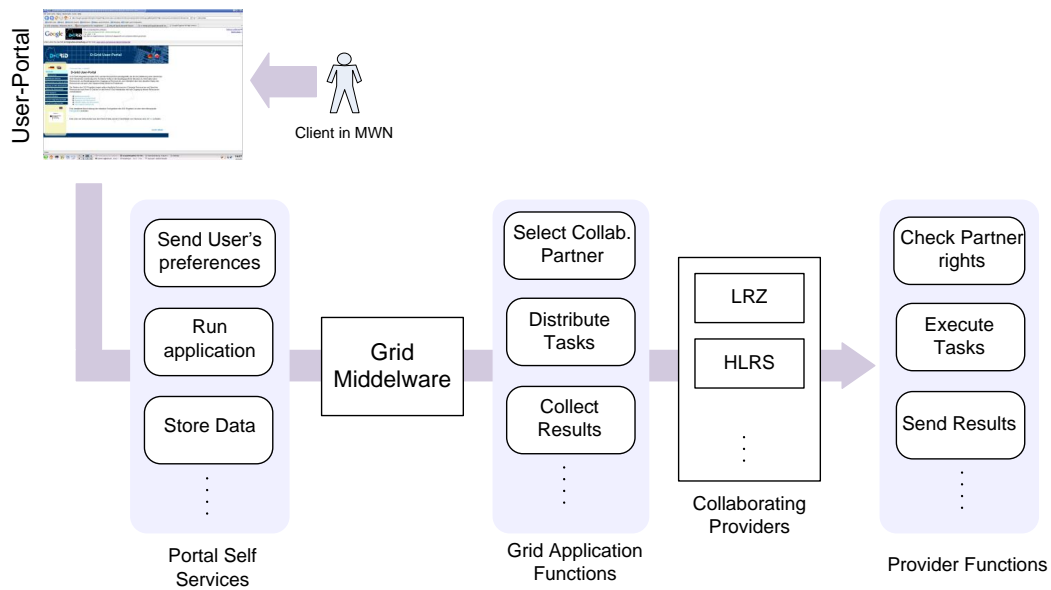


Figure 2.20: Generalised application scenario for a Grid environment

the requested task will then be sent to the selected provider, according to several organizational as well as technical criteria.

- As soon as the task has been executed on the provider side, the result will be aggregated and sent back to the client.

Grid environment in comparison with the CoT

In this context, the set of *Real organizations* (ROs) (represented by the service providers such as the LRZ as an autonomous organization) with the common purpose or interest of sharing their resources to further their objectives build up the so-called *Virtual Organization* (VO). Several responsibilities for the management of the Grid environment are, subsequently, delegated to the VO, which in terms of CoT plays the role of the CoT-Founder.

In the following table (Table 2.4) we shall check the characteristics of the Grid environment against those formal definitions of the CoT:

Circle of Trust (CoT)	Grid environment
CoT Principals	RO such as providers of services and resources for storage or computational purposes (for example the supercomputing centres in the case of DEISA project) as well as the end users represent the main principal groups in the Grid environment.
CoT Roles	CoT members are the ROs and CoT Founder is enforced by the virtual organization concept.
CoT Agreements	Defined in the Service Level Agreements among the ROs, where the participants agree on conditions and rules for sharing resources and using services.
CoT Trust Relationships	Supported by the authentication and authorization federation aspects (federated authorization through SAML assertion [SAM03]).
CoT Shared resources	The resources are typically computers, data, software, expertise, sensors, instruments, etc.
CoT Platform	The technical platform of Grid is typically realized in the Grid-Middleware which is actually based on Web services as the main communication platform.
CoT Protocols	Web services protocols as well as other security protocols such as SAML [SAM03].

Table 2.4: Grid environment characteristics in light of the formal definition of the CoT

2.2.3.2 Actual authorization and security models in Grid Computing

Grid computing is seen as the upcoming technology for solving complex computational problems to make possible the sharing of services distributed across multiple organizations. The real organizations linked in the Grid, forming the virtual computational space, might have different policies for the management of resources. Based on that, it is obvious, that when it comes to service integration across multiple partners, both security and trust issues may not be neglected.

With respect to security, most known Grid software focuses primarily on authentication, access control and ease of collaboration. Authentication basically identifies each participant and ensure that no unauthorized parties are involved, while access control ensures that the participant is allowed to use the resources and services offered by remote participants. Currently Grid security uses X.509-based digital certificates [X50], security assertions (SAML) [SAM03] or role-based access management solution, such as PERMIS [PER] and Shibboleth [X50].

However, beside these security aspects, the overall decision whether to rely at all on a collaboration partner or not, may be affected by other non-functional aspects that cannot be generally determined for every possible situation, but should rather be under the control of additional aspects, such as aspects related to the reputation of the partner and past behavior. The next subsection will deal with these aspects in more detail.

2.2.3.3 Problems and open issues

As we stated earlier, the basic idea behind the concept of Grid environment evolves around the concept of Virtual Organizations (VOs), which enable different organizations or individuals to share resources in a controlled fashion to achieve a common goal (the organizations agree on common conditions and rules for sharing resources and using services).

Beside this static constellation of VO, in several real world scenarios (such as the case of Mamogrid Medical Image and Video Analysis [AEH⁺04] or in the field of the areas of physics, chemistry, and biology for scientific simulations [Fer05]) the participants may organize themselves, *on the fly*, into a group and thus form a dynamic VO (DVO) [FKNT02]. In doing so, any new VO can be made available and offer its functionalities to every other participant in the environment. The feature of DVO is actually the key behind higher robustness and lower costs for the management of Grid systems.

However, this feature facilitates the deployment of new services, but at the same time raises many problems, from performance degradation to the growing of the uncertainty in the environment as the number of participants grows.

Further, regarding the available security protection mechanisms (where the only trust notion contained there is represented by the *trusted CAs list*, in which all CAs that a participant trusts are listed) the participants can not be confident that applications that run on the remote sites, either on behalf of consumers or providers, are going to behave properly. Additionally, if there are behavioral deviations, then it is also not clear under what circumstances the deviating behavior of a partner is going to be tolerated.

Similar to the DL scenario, trust, in this regard, can be seen from two angles: (i) *Collaboration Trust*, and (ii) *Quality of Service Trust*.

Collaboration Trust

Collaboration trust is concerned with the trustworthiness of an interaction partner. This type of trust is important to investigate because uncertainty regarding trusting other partners predominates in Grid environments. It is closely connected with the lack of information about the participants and especially about their behaviors.

This uncertainty is based on the fact that a single job may access resources with many different owners with different trust requirements. The domain managers in turn cannot be expected to obtain and keep track of all the associated certificates and private keys. Therefore, they need software to automate the process. Nor can resource owners have a local account in place for every potential user or keep track of all the relevant changes in users' qualifications, such as group memberships, etc.

The following categories for uncertainties in Grids are identified:

- *Uncertainty on current behavior*; this issue is the result of the absence of tools for monitoring the ongoing collaboration between participants, for verifying the outcoming cooperation results with the set of rules and preferences established by the VO.
- *Uncertainty on future behavior*; uncertainty in Grid environments is a severe problem to deal with, because the absence of a history regarding the past behavior of the collaboration parties makes it also difficult to reason and create a logic on

possible future behavior.

Quality of Service Trust

Beside Collaboration Trust, Quality of Service Trust considers issues on trusting the quality of the resource and/or the service provided by the cooperation partner. In these environments, the participants cannot be aware of the nature of its collaboration parties or of the quality of their resources and services. Therefore, QoS properties can be considered as trust indicators because if their specific values at a certain moment of time do not fit with the given and agreed upon quality insurance rules, this may considerably influence trust among the involved partners.

2.2.3.4 Prospective solution

A prospective solution for the above mentioned problems needs first and foremost an approach, which can be based on statistical methods of quality assurance, so that decisions on *actual* interactions based on the knowledge derived from *observed* interactions can be taken automatically.

Collaboration Trust

- When trusting a participant, it is important to know which aspect one is referring to. There are instances where a participant is trusted more than the others regarding different situations. In this solution, there must be the possibility to specify in which aspect of trust participants are interested in and at which level. Accordingly, trust towards a participant should be handled in different contexts. Further, these contexts should be used to decide whether a participant is eligible for a certain action or not.
- Subsequently, trust may be derived from a multitude of aspects and dimensions. Therefore, an overall trust value of a participant should be represented in a unified manner, so that other participants can automate the process of access decision more easily.
- An additional alternative solution for reducing uncertainty is to reason about past experiences. The goal is to monitor the progress of the collaboration between participants, since the definition of policies regarding the collaboration does not guarantee a secure collaboration; therefore monitoring plays therefore a crucial role. This is essential to ensure that the interaction is progressing according to the needs and preferences of the collaborating parties leaving less space for surprises regarding the outcome of this interaction.
- Possible output of such a monitoring process may help to create a history of the behavior of all collaboration partners.

Quality of Service Trust

An alternative solution, with regard to the quality concerns, should give the providers the possibility to specify explicit QoS assurances regarding, for example, availability, stability, and capability.

Based on that, extracting trust from the QoS parameters can serve as a criterion for the involved parties when selecting appropriate partner cooperations. This feature will also

serve to push participants to continuously improve their behavior and the QoS of the services they offer. In the opposite case, mechanisms for protecting the resources with risk parameters are required as well.

2.2.3.5 Requirements from the DEISA Project scenario

Having looked at the problems facing the organizations in grid computing environments, and the reason why investigations on a trust model are fundamental for these issues, we now move on to look at the requirements for the realization of such a trust management solution to support collaboration across external boundaries.

Similar to the digital Library scenario, trust assessment tends to be either related to the participants' behaviors and identities when they play the role of service consumers, or it can be related to the quality of the services/resources they provide, when they have the role of service providers.

Accordingly, the range of requirements tends to fall into the well-known categories: Collaboration Trust and Quality of Service Trust.

2.2.3.5.1 Collaboration Trust For collaboration trust, most of the requirements we introduced in Subsection 2.2.1.4 apply in the same sequence:

Security management requirements

Security is a crucial aspect in Grids, where basically most of the classical security aspects such as authentication, access control and authorization (AAA) need to be deployed globally among the participants within the Grids. Therefore, the requirement [SEC-AAA] applies to this case study.

Besides, as we discussed earlier, VOs are formed dynamically where any member can join and leave anytime and anywhere. As the members are from different security domains, they may not share the same security policy. Consequently, the requirement [SEC-Policy] for global security requirements plays an important role as well.

Trust level assessment requirements

From the scenario given above, we provided reasons why identity certificates and local accounts alone cannot be the basis for authorization decisions in a large-scale Grid, and that a proper trust evaluation model for grid is needed.

For this purpose, all the requirements deduced from the IntegraTUM scenario in Subsection 2.2.1, especially for the computation and the representation of trust values among the participants ([Trust-Intermediary], [Trust-Level], [Trust-Metric], [Trust-Context] and [Trust-Policy]) need to be considered as well. This includes defining direct or mutual trust relationships between two hosts within a domain, as well as indirect trust relationships when traversing intermediaries.

In the context of trust estimation, the requirements for estimating trust from the well-known aspects are as follows:

- **Delegation:** The practice today in Grids is for the job to authenticate to all services with its single proxy certificate, delegated to it on initial submission. For example, owners who provide resources that are part of a large Grid

will often be willing to give up authority to decide who is entitled to access those resources. This circumstance induces the fulfilment of the requirements [Delegation-authorization] and [Delegation-TTP-Trust].

- **Past experience and auditing requirements:** On today's Grids estimating trust from past experience and auditing mechanisms is gaining more and more attention. Therefore, the related requirements on this matter (Audit-Information), [Audit-Evaluation] and [Audit-Storage]) need to be taken into consideration as well.
- **Reputation management requirements:** Similarly, all the requirements on reputation management ([REP-Value], [REP-Metric], [REP-Context], [REP-Credibility] and [REP-Recentness]), which give the participants the possibility to rate the performance of each other's behavior, service or resources apply for the realization of our trust management solution.

Obviously, for the final assignment of trust values to principals, the different trust values that result from varying mechanisms have to be consolidated and aggregated. Accordingly, the trust aggregation requirements [Aggre-Collection] and [Aggre-Scheme], discussed in Subsection 2.2.2.6, are appropriate therefore.

Interorganizational access control requirements

In today's Grids, the size and complexity of the virtual organizations they can support is limited by the burden placed on resource managers to manage privileges based on the identity of each user in the virtual organization.

To address this scaling issue, interorganizational access control methods, which cover solutions on access authority to access information distributed storage ([Access-Authority], [Access-Policy] and [Access-Storage]), need to be enhanced.

Organizational Requirements

By definition, the concept of virtual organizations in Grids reflects most of the characteristics of the CoT. For this reason, the well-known organizational requirements ([ORG-TLA], [ORG-Time], [ORG-Simplicity], [ORG-Cost], [ORG-Integrity] and [ORG-Impact]), for example for setting the TLAs among the participants in the VO or for ensuring the integrity of the VO, may not be disregarded.

Privacy requirements

The same argumentation holds for the requirements on privacy ([Priv-Collection] and [Priv-Use]) in relation with data and resource sharing between the VO members.

Technical realization requirements

All of the requirements mentioned above need to be realized on top of the existing technical platform and communication protocols in use in the VO management interfaces. Based on that, the technical realization requirements ([Technical-Integrity], [Technical-Protocols] and [Technical-Storage]) for cross domain operations are needed.

2.2.3.5.2 Quality of Service Trust In analogy with content quality trust in the DL scenario (see Section 2.2.2), Quality of Service Trust addresses the influences that determine trust with regard to the quality of services and resources provided in the VO.

Quality of Service Trust requirements

Exactly like the requirement [Content-Parameter-Trust], the requirement [QoS-Trust] for the provision of QoS information (usually classified in distinguished parameters within a standardized scheme) is beneficial to develop trust in the service provider therefrom.

Risk management requirements

In addition to the QoS parameters, which allow to reason about trusting the service provider, this reasoning in Grids entails an additional bulk of requirements that deal with risk management in order to balance the costs of trust betrayal. This is because one price of engaging in consequential relationships at a distance and putting trust in another partner creates vulnerability.

In the following we enumerate some relevant requirements for associating trust with risk:

- **Risk Level:** In the context of risk management, an important aspect of attributing levels of trust to principals for distinct actions is also the *risk*, to which trust must be balanced. Consequently, the shared services and resources in Grids need to be defined with risk factors making fine-grained access decisions with all the information available to the resource owner at the time of the decision.

In doing so, the resource owner will have the opportunity to indicate a level of importance of the resource as well as the possibility to set up certain risk information on the given action, such as it can be challenged, for example, that reduced risk and increased trust may both increase the likelihood of approving the access control for this action, and engaging in a cooperation with the unknown entity [Risk-Level].

- **Risk Metric:** Similar to the requirements [Trust-Metric] and [REP-Metric], a metric for representing the risk information such as a degree of risk and other alternative risk attributes need to be defined uniformly in the CoT [Risk-Metric].
- **Balance between trust and risk:** As discussed earlier, the trust in an unknown principal must be balanced against the risk level set to a certain resource in general or to an action on the given resource in particular.

Within that scope, rules and conditions for expressing the way risk assessment can be included in the trust-based decision-making approach need to be investigated. This investigation should cover several sorts of situations, especially when the trustworthiness of some entity cannot be estimated because, for example no recommendation information is available at the time of the decision [Risk-Rule].

Technical realization requirements

The technical realization of the requirements that are related to Quality of Service Trust engenders other requirements regarding the storage complexity ([Storage-Complexity]), the monitoring of this content ([Storage-Monitoring]) as well as eventual conflicts between collaboration and quality of Service Trust ([Storage-Conflict]).

2.2.4 Conclusion: Need of a generic model of CoT

In the course of the previous sections, we have demonstrated by means of three illustrative scenarios, which concretize the three different classes of the CoT, that organizations have to face many challenging problems in resource sharing and collaborative environments. These challenges involve trust in unknown entities, access control, risk aspects as well as the dynamic nature of the entities' behavior and content.

Further, we provided conception ideas as well as requirements on prospective solutions that evolve around designing a generic model for extending the actual models of CoT with a more generic trust assessment model. The main goals of the generic model of CoT are basically consolidating trust relationships, categorising access rights accordingly and introducing automation through the lifecycle of the trust relationships.

However, as we will discuss in the next section, the changing nature of FEs requires the design of an appropriate change management process in accordance with the life cycle of the CoT. The change management process on the one hand needs to support the processing of eventual changes, and on the other hand it should enable traceability of these changes, which should be possible through proper execution of the process described.

2.3 Use Cases for the management of CoT

The CoT at every level can be affected by changes. Trustworthiness of principals as well as resource description – as defined in the previous sections – may change over a period of time. That is, there is an issue over the control of this changing data - information for estimating trust having been altered may in fact be retained for access control decisions or passed over to third parties.

This issue is highlighted in order to convey the necessity of trust change management solution that improves the ability to cope with the change brought about principals, services and resources. Below, we quote relevant requirements on a change management process in the CoT:

2.3.1 Requirements for the extension of the CoT with a change management process

- **Dynamic change:** As we stated in the previous sections, the federated environment might be highly dynamic, because the joint organizations may participate in multiple collaborative environments based on different needs and contexts of collaboration. Due to this fact, the membership of the organizations in the resource sharing process can be short-lived and may constantly change. Additionally, the users roles and responsibilities in service usage may correspondingly change; and thus these dynamic changes may involve update requirements on the security policies as well as privacy constraints.

We conclude that the FE cannot be static and tightly coupled to applications. It must be adaptive to account for these changes without having to modify radically the existing organization security infrastructures [Sec-Update].

- **Trust Level Update** The computed trust level for entities involved in the CoT, may change over time to reflect the increase or the decrease in the trust. As a result, the degree of trust in the entity needs to be constantly re-evaluated based on freshly obtained information or interaction feedbacks accumulated with time or communicated from other trusted entities [Trust-Update].
- **Reputation Update**: Similarly, the nature of the entity's behaviors per interaction, the reputation values as well as the evaluations may naturally change as well. There is, therefore, a need of a dynamic update of the reputation values. This update has to take the entire history as well as the type of the interactions into account [Rep-Update].
- **Risk Update**: Information about the current resources being shared in the FE play an important role; since this information may be assigned to roles and privileges. Thus, it is necessary to provide resource owners with the possibility to reevaluate the risk parameters bound to the resources provided in the CoT [Risk-Update].
- **Change Notifications**: One of the important security measures that have to be considered in adopting trust management for the problems stated above, is to keep the resource owner notified about even minor changes to their shared resources. For example in DL scenario, the authors need be warned of modifications to their articles, allowing them to verify the validity of new editions quickly [Change-Notify].

2.4 Assessment of the requirements

The presented requirements are gathered from a number of scenarios and use cases including considerations from representation matters of the trust data, from technical realization aspects as well as from organizational agreements impacts.

In accordance with the prospective solutions extracted from the given scenarios, we shall in the following elaborate a weighting of these requirements and classify them by order of priority.

2.4.1 Classification and weighting of the requirements

After analyzing the requirements for realizing a trust management solution, we need to take a few more steps, as these requirements must be prioritized. For this purpose, we will be applying weights to the requirements in order to give them scoring importance in proportion to their importance in supporting the functions of the potential solution.

We shall use a general three-level prioritization scale:

- (2) This priority indicates a requirement that is *essential* and that the CoT must have to be able to manage the trust relationships.
- (1) indicates a requirement which is *important* for the CoT, and which adds significant functionality to the trust management solution.

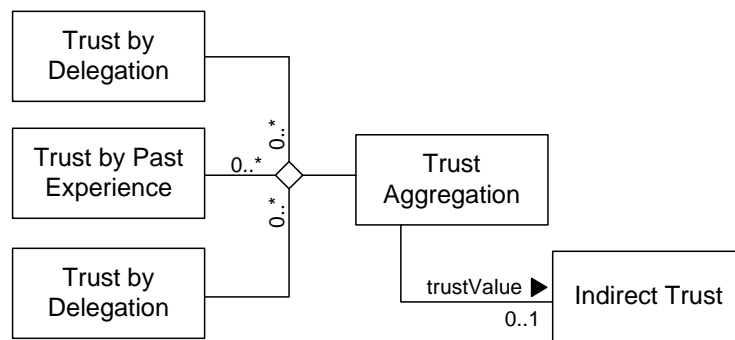


Figure 2.21: Dependencies for Indirect Trust

(0) indicates a requirement that is *nice to have* (informative) but not essential.

Below we shall recapitulate all the requirements groups, discussed in Subsection 2.2.1.4 following the same order and dependencies, and justify the attributed weight of each requirement.

For the classification of the requirements on indirect trust, we follow the same argumentation given in Subsection 2.1.2.1, when we discussed that indirect trust can be derived from three distinct dimensions: Delegation, past experiences as well as reputation information.

Figure 2.21 shows the way these aspects are associated to each other. This association can be handled by means of the *Aggregation Trust* class, which in turn, is associated with the final *indirect trust* class for representing the final trust levels. However, this association means that when at least one of the given dimensions exists, either one aggregated information can be provided to the indirect trust class or no such information is available.

The weighting of the requirements within each of these classes will be represented in the following tables:

Direct Trust	Weight	Summary and justification (see page (41))
[SEC-Policy]	(1)	<p>Summary: In an environment with a large number of organizations with many separate security requirements, it is useful to have a set of baselines that define global security policies among them.</p> <p>Justification: Not only do these baselines ease the burden of updates for policies specifiers in the long term, it also prevents potential conflicts regarding trust based access control for unknown entities.</p>
[SEC-AAA]	(1)	<p>Summary: Represent the need of global security policies with regards to the AAA mechanisms.</p> <p>Justification: Obviously, this requirement belongs to the agreements that could be made among the organizations for setting global policies and therefore it has the same priority level as the previous requirement.</p>

Indirect trust (in general)	Weight	Summary and justification (see page (42))
[Trust-Interm]	(2)	<p>Summary: In the given definition of indirect trust, we argued that indirect trust in most of its dimensions rely on recommendations from intermediaries (TTP). These intermediaries are entities which facilitate interactions between two parties, basically under the condition that both parties trust the third party. This is a common case for TTP in cryptographic protocols, for example, a certificate authority (CA).</p> <p>Justification: There is the issue of the two parties being able to properly identify the recommender as a trustful entity. Therefore, this requirement is <i>essential</i> because when the trustworthiness of the intermediaries is not verified, indirect trust can not be established appropriately.</p>
[Trust-Level]	(2)	<p>Summary: It represent the quantification of trust into different levels.</p> <p>Justification: This requirement is essential, because, in this work, we are dealing with trust to support a decision through the use of trust-related thresholds that along with trust values may be used for the access reasoning. Consequently, the automation of access decision process among the members of the CoT would be impossible in the absence of unified trust levels.</p>
[Trust-Metric]	(2)	<p>Summary: It implies the design of a unified metric for representing the trust levels.</p>

Justification: Following the importance argumentation of the previous requirement [Trust-Level], we consider this requirement to be essential as well. This argumentation is based on the fact that the trust management solution can not be standardized in the CoT insofar the trust reasoning algorithms rely on a common metric for the representation and interpretation of the trust levels.

[Trust-Context] (2) **Summary:** This requirement considers the context in which trust can be estimated, thus enables a refinement of the decision making.

Justification: Without a reference to each specific context, the trust level can merely be assigned to an entity for a general use. Obviously this may lead to erroneous results.

[Trust-Policy] (2) **Summary:** It addresses the need of a unified set of statements and rules that express how much trust is needed for performing a given action.

Justification: It is apparent that this requirement is just as essential, because the trust reasoning process can not be automated without the use of these rules.

Indirect trust (by delegation)	Weight	Summary and justification (see page (44))
--------------------------------	--------	---

[Deleg-Auth]	(1)	Summary: This requirement claims that indirect trust can be derived from delegated authorities in a way that a principal <i>A</i> trusts principal <i>B</i> more or less relatively to authority delegation from principal <i>C</i> .
--------------	-----	--

Justification: Accordingly, this requirement is *important*, especially in situations where no other information about the unknown entity is available.

[Deleg-TTP]	(2)	Summary: Similar to the requirement [Trust-Interm], this requirement considers the trust in the entity which delegates the rights (principal <i>C</i>).
-------------	-----	---

Justification: Delegation is normally necessary for building trust, but without this requirement, is not sufficient especially when, for example, a threshold under which the trust level of the delegator is not enough for delegating or there might be other reasons preventing delegation.

Indirect trust (past experience)	Weight	Summary and justification (see page (60))
----------------------------------	--------	---

[Audit-Info]	(1)	Summary: In analogy with indirect trust by delegation, this requirement deals with building trust from past behaviors and experiences.
--------------	-----	---

Justification: We argue that this requirement is important, because the trust model needs to evaluate the information that is extracted from past experiences, otherwise, the model is faced with issues of mistaken or altering audit data.

[Audit-Eval] (1) **Summary:** In the context of trust building, the audit trails must be processed to provide security administrators with only the information of interest rather than series of data.

Justification: Without a unified representation of the audit records, it will not be possible to automate this task among the different security domains.

[Audit-Stor] (1) **Summary:** This requirement deals with storing the audit data from multiple domains.

Justification: The automation of this process is not possible, when issues on the expense of storing the audit trails, as well as matters on distributed access to this data are not investigated.

Indirect trust (by reputation)	Weight	Summary and justification (see page (60))
--------------------------------	--------	---

[REP-Value]	(1)	Summary: It represents the reputation information (ratings), as an additional information source that might affect trust, into reputation values.
-------------	-----	--

Justification: In instances when there is no personal experience with the requester as well as no delegation authority from a trusted entity, the reputation information gathered from other members in the CoT might be the only source for estimating the trustworthiness of an unknown entity.

[REP-Metric]	(1)	Summary: Here a standard metric for representing the rating information is required for making this information useful for trust building in the CoT.
--------------	-----	--

Justification: In the absence of such a metric, the CoT members would be faced with heterogeneous syntactic/semantic reputation data representation.

[REP-Context]	(1)	Summary: Similar to the requirement [Trust-Context], this requirement specifies the context of the reputation value.
---------------	-----	---

Justification: The context of the reputation values is important in a sense that otherwise these values would merely state whether the principal has a good or bad reputation in general. Obviously this may lead to conflicts, when the reputation of the principal considerably changes over multiple contexts of the interaction.

[REP-Cred]	(1)	Summary: Requirement on the credibility of the rating plays an important role to help differentiate between honest and dishonest ratings.
------------	-----	--

Justification: In this regard, this requirement is important to prevent trust to be build from malicious ratings.

[REP-Recent] (1) **Summary:** It considers the recentness factor of the rating of each contributor, and privileges the most recent ones.

Justification: In the same argumentation, the aspect of recentness of the ratings prevents forming trust from information that may be out-of-date.

Trust Aggregation	Weight	Summary and justification (see page (62))
[Aggre-Collect]	(2)	<p>Summary: In the case the trust values are computed from different aspects (delegation, past experience and reputation), the requirement [Aggre-Collect] must be applied for collecting this information with a tight reference to the corresponding dimension.</p> <p>Justification: This requirement is essential for ensuring an automatic mapping between trust values and dimensions and in that way it prevents erroneous results.</p>
[Aggre-Scheme]	(2)	<p>Summary: By means of an aggregation scheme, this requirement allows a standard way for aggregating the trust values in the CoT.</p> <p>Justification: In the case where a general trust value about the principal is required for an access decision, this requirement facilitates the aggregation task and additionally relieves the members from this charge.</p>

To recapitulate the relationships between the requirement classes we discussed so far, we recall the representation of the dependencies between the classes of the trust dimensions (as shown in Figure 2.21), and extend it with the associated weighted requirements as illustrated in Figure 2.22.

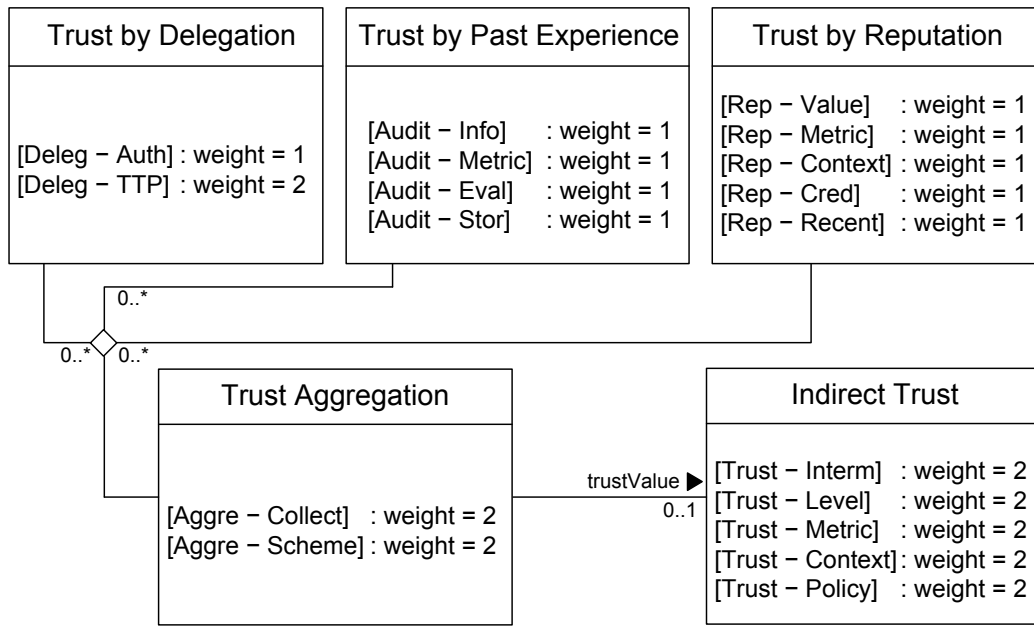


Figure 2.22: Dependencies and weighting of the requirements leading to indirect trust

Interorganizational access control	Weight	Summary and justification (see page (44))
[Access-Auth]	(1)	<p>Summary: It gives each organization the possibility to preserve and to control the authority over the resources being shared in the CoT.</p> <p>Justification: This requirement is important because disregarding the organization’s desire to retain some authority over the resources may influence the collaboration of the organization in context of the CoT.</p>
[Access-Policy]	(2)	<p>Summary: It implies the establishment of distributed access policies, which actually represent one of the major aspects for creating a CoT among a set of organizations.</p> <p>Justification: This requirement is seen as essential due to the expected risk when the organization providing the resource does not export its access control policies in a form that can be understood by other members, so that unknown entities (who are trying to gain access to the local resources of interest to them) understand the purpose of the requirements.</p>
[Access-Stor]	(1)	<p>Summary: In this requirement, the storage of the authorization information, which can serve as a history record for future interactions with the same entity is required.</p>

Justification: This requirement is relevant for building trust from past experiences, particularly in instances where reputations as well as delegation information are not available for reasoning about the requester's trustworthiness.

Technical realization	Weight	Summary and justification (see page (47))
[Tech-Integr.]	(2)	<p>Summary: This requirement regards the safeguarding of the CoT integrity. This is because the technical realization of the trust building mechanism may require the development of new architectural components and thus may extend the CoT.</p> <p>Justification: It has a priority level 2 (essential), because this extension may not have a lasting effect on the integrity of the CoT infrastructure, otherwise the whole concept of the CoT may be consequently affected.</p>
[Tech-Protocol]	(1)	<p>Summary: It introduces the need for trust negotiation protocols as a means of communication with the requester.</p> <p>Justification: In case these protocols are missing in the CoT or are not appropriate for this aim, this requirement would then be important to fulfill this need.</p>
[Tech-Storage]	(1)	<p>Summary: Shared storage techniques with standardized schemes should be available for representing the trust data (e.g. reputation data) along with its metric.</p> <p>Justification: For the purpose to more easily eliminate duplicates and avoid inconsistencies in the stored trust data, this requirement is regarded as important.</p>

Organizational requirements	Weight	Summary and justification (see page (45))
[ORG-TLA]	(1)	<p>Summary: In the formal definition of the CoT, the concept of the Trust Level Agreements (TLAs) among the CoT-members play a prominent role.</p> <p>Justification: The importance of this requirement is enforced by the fact, that it allows the members to express their guidelines and policies such as access control and privacy policies in a standardized manner.</p>
[ORG-Time]	(0)	<p>Summary: Due to the dynamic nature of the prospective access request, making access decision in real time is helpful for a quick setup of the requested cooperation.</p> <p>Justification: The feature of this requirement is considered as nice to have because it is not essential for the setup of the requested cooperation.</p>

[ORG-Simple]	(0)	Summary: The simplicity of the description of TLAs improves a common understanding of the CoT guidelines.
Justification: In the same argumentation as the previous requirement, this feature is nice to have but not essential.		
[ORG-Cost]	(0)	Summary: Obviously, it is advantageous when the setup of the TLAs can be kept as low as possible, in order to reflect the benefit for the collaborations in the CoT.
Justification: The setup of the TLAs can also be costly and this is the reason why this requirement is rated as nice to have.		
[ORG-Integr]	(1)	Summary: As stated earlier, these two requirements aim at preserving the organizational integrity of the CoT as well as the integrity of other relationships across the borders of the CoT.
[ORG-Impact]	(1)	
Justification: These requirements are important to ascertain that the temporal inclusion of unknown entities therein may not either affect the CoT integrity nor have an impact on other third parties relationships.		

Privacy	Weight	Summary and justification (see page (46))
[Priv-Collect]	(1)	Summary: This requirement means that any member in the CoT processing the trust information data, usually collected across domains, must comply with the privacy key principles about this data.
Justification: In the absence of these principles and constraints, there is a risk that the trust data will be stored or read by any other third parties without consent. For this reason, this requirement is regarded as important.		
[Priv-Use]	(1)	Summary: This requirement concerns issues that evolve around the way the trust data may be handled and shared in the CoT.
Justification: It is important, because it aims at maintaining security for participants by ensuring that the members have the ability to share only the data they need, and no more.		

Risk Management	Weight	Summary and justification (see page (72))
[Risk-Level]	(1)	<p>Summary: The risk information needs to be represented in risk levels especially for situations of uncertainty in which the likelihood and consequences of a particular risk on a given resource might be critical.</p> <p>Justification: It is important because if access decisions were merely based on trust information the risk of uncertainty might be increased.</p>
[Risk-Metric]	(1)	<p>Summary: It represents the need of a standardized metric for representing risk information in the CoT.</p> <p>Justification: Following the same arguments given for the requirements [Trust-Metric] and [Rep-Metric], it is obviously not possible without such a risk metric to reason about the risk information in an automatic and consistent manner.</p>
[Risk-Rule]	(1)	<p>Summary: The risk level information is consequently needed in dedicated rules, which reason about trust against risk information for performing an action on a specified resource.</p> <p>Justification: Without these statements and rules the trust management solution can not make use of the risk information with regard to trust.</p>

Change Management	Weight	Summary and justification (see page (73))
[Sec-Update] [Trust-Update] [Rep-Update] [Risk-Update]	(2) (2) (2) (2)	<p>Summary: The input information for the trust reasoning, which is typically based on the privacy policies, the trust levels from its different dimensions, the reputation values, as well as the risk information need to be updated regularly.</p> <p>Justification: Obviously this requirement is essential in a sense that the access decision relies on this input information. Consequently, the accuracy of the obtained trust judgments can only be ensured when this data is kept up-to-date and faultless.</p>
[Notify]	(0)	<p>Summary: On the basis of the definitions given in this chapter, trust concept in any relationship will inevitably change (increase or decrease). Accordingly, changes in this context subsequently lead to changes with respect to the resource description, the privacy policies, etc.</p> <p>Justification: Due to the fact that these changes may affect the resources being federated in the CoT, it would be helpful to notify the entities concerned by these changes regularly.</p>

Content Quality Trust	Weight	Summary and justification (see page (64) and page (71))
[Content-Quality]	(1)	<p>Summary: In resource sharing scenarios such as document content sharing scenarios, the quality of the content being shared plays an important role for building trust of these resources' issuer.</p> <p>Justification: Here again a standardized metric or a scale for describing the content quality parameters is needed, without which building trust through content quality can not be automated.</p>
[Content-Rep]	(0)	<p>Summary: As we discussed in page (64), exchanging rating feedback about the content of the resources can enforce the notion of trust by content quality.</p> <p>Justification: This feature is nice to have but not essential, because content trust can also be estimated without the existence of these ratings.</p>
[Store-Complex] [Store-Monitor]	(0) (1)	<p>Summary: These requirements aim at helping the organization members to tackle trust data storage complexity and perform quality checks on the trust data, and thus fasten the computation as well as the update of the trust information.</p> <p>Justification: These features are also nice to have rather than important, because these performance aspects depend on the capacity of the members infrastructure. In the same context, operating data quality checks that should be conform with the monitor systems in the CoT are helpful to manage instantaneous notifications and warnings.</p>
[Store-Conflict]	(1)	<p>Summary: Attributing trust values to the content quality may lead to conflicting results about the trustworthiness of identities against that of resources' content.</p> <p>Justification: Obviously, neglecting this requirement may produce erroneous results.</p>

2.4.2 Summarization - Criteria catalogue

In the following table 2.23, a criteria catalogue that sums up all the requirements as well as their weights for our trust management solution will be used as means to identify how the aggregated requirements can be fulfilled, and will serve as an input into the design stage of our solution in the next chapters.

Direct Trust Requirements		Indirect Trust Requirements	
[SEC - AAA]	(1)	[Trust - Intern]	(2)
[SEC - Policy]	(1)	[Trust - Level]	(2)
		[Trust - Metric]	(2)
		[Trust - Context]	(2)
		[Trust - Policy]	(2)
Trust by Delegation		Trust by past experience	Trust by reputation
[Deleg - Auth]	(1)	[Audit - Info]	(1)
[Deleg - TTP]	(2)	[Audit - Eval]	(1)
		[Audit - Stor]	(1)
			[Rep - Value]
			(1)
			[Rep - Metric]
			(1)
			[Rep - Context]
			(1)
			[Rep - Cred]
			(1)
			[Rep - Recent]
			(1)
Trust Aggregation			
[Aggre - Collect]	(2)	[Aggre - Scheme]	(2)
Interorganizational Access Control Requirements		Technical Realization Requirements	
[Access - Auth]	(1)	[Tech - Integrity]	(2)
[Access - Policy]	(2)	[Tech - Protocol]	(1)
[Access - Stor]	(1)	[Tech - Storage]	(1)
Organizational Requirements			
[ORG - TLA]	(1)	[ORG - Cost]	(0)
[ORG - Time]	(0)	[ORG - Integr]	(1)
[ORG - Simple]	(0)	[ORG - Impact]	(1)
Policy Control Requirements			
Privacy Management		Risk Management	
[Priv - Collect]	(1)	[Risk - Level]	(1)
[Priv - Use]	(1)	[Risk - Metric]	(1)
		[Risk - Rule]	(1)
Change Management Requirements			
[Sec - Update]	(2)	[Risk - Update]	(2)
[Trust - Update]	(2)	[Notify]	(0)
[Rep - Update]	(2)		
Content Quality Trust			
[Content - Quality]	(1)	[Store - Complex]	(0)
[Content - Rep]	(0)	[Store - Monitor]	(0)
		[Store - Conflict]	(1)

Figure 2.23: Criteria catalogue

Chapter 3

Related Works in Trust Management and Access Control

"One must be fond of people and trust them if one is not to make a mess of life."

E.M. Forster

Contents

3.1 Trust definitions	89
3.1.1 Trust establishment and trust relationships	89
3.1.2 Circle of Trust (Liberty Alliance Project)	96
3.2 Indirect trust dimensions	100
3.2.1 Indirect trust by delegation	100
3.2.2 Indirect trust from past experience	101
3.2.3 Indirect trust by reputation	104
3.2.4 Indirect trust aggregation	106
3.2.5 Fulfillment of the requirements?	107
3.3 Interorganizational access control mechanisms	107
3.3.1 Intraorganizational access control models	108
3.3.2 Extension tentatives to interorganizational scenarios	109
3.3.3 Shortcomings and fulfillment of the requirements	109
3.4 Policy control	110
3.4.1 Privacy management	110
3.4.2 Risk management	111
3.5 Organizational Trust	112
3.5.1 Defining Trust by law	112
3.5.2 Discussion	113
3.6 Content quality trust	113
3.6.1 Wikipedia Case Study	114
3.6.2 Shortcomings and fulfillment of the requirements	116
3.7 Prototypes – Solutions for automated trust assessment	116

3.7.1	PolicyMaker and KeyNote	117
3.7.2	Trust Policy Language (TPL)	117
3.7.3	REFEREE Trust Management Model	118
3.7.4	Standards for the World Wide Web	118
3.7.5	Shortcomings of these automated trust assessment systems	119
3.8	Analysis and conclusions	120
3.8.1	Discussions	120
3.8.2	Update of the criteria catalogue	121

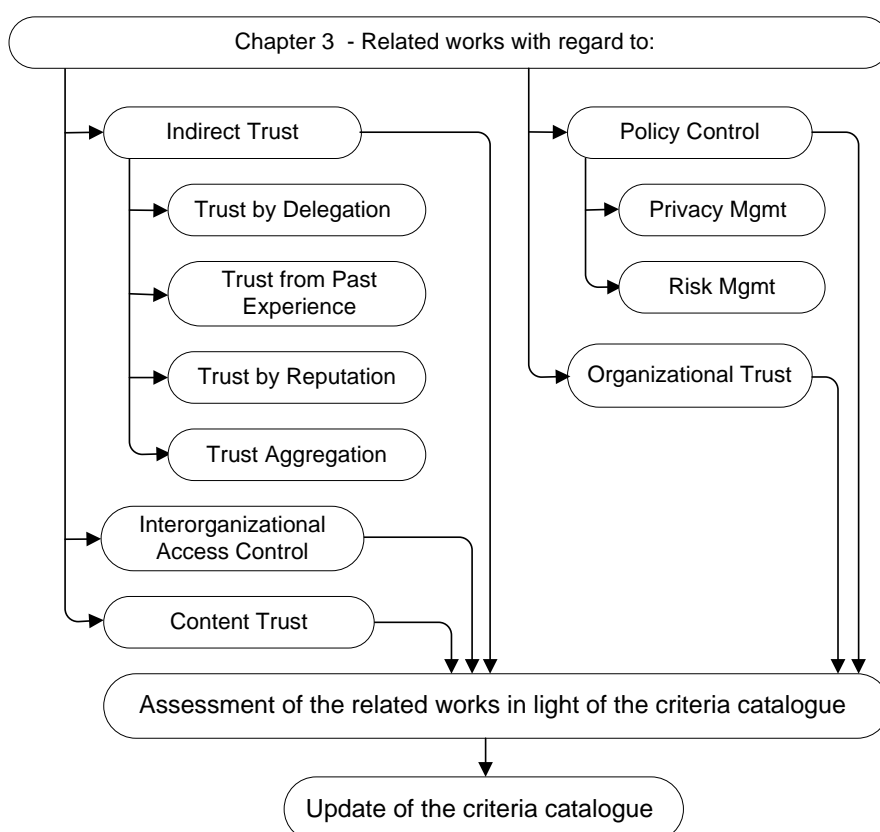


Figure 3.1: Sequence structure for chapter 3

Several different approaches for the management of trust, access policies, and authorization have been explored in distributed and federated environments. A good overview is given by [GS00]. However, no single unified definition of trust does exist in the current computer science literature.

As we will discuss in the course of this chapter, most of the definitions found in these related works classify trust as a number, as a discrete labeled degree, as a flag or as a combination of all of these.

In this context there are also various ways to define and establish a trust relationship. In some scenarios, the trust relationship can be negotiated if the collaboration is among

real organizations. It can be specified as an eContract in XML [GBW⁺98] and later can be exchanged and modified by collaborating organizations.

In other scenarios, the trust agreement can be specified by one party (e.g. a service provider) and accepted by another party without negotiation (e.g. a consumer of the service). However, another example of trust establishment implies that the trust agreement can be declared by a controlling entity and be applied by all the involved parties (e.g. global policies declared by the founder of the federated environment).

Actually, one of the first works that tried to give a formal management of trust, that could be used in computer science, was that of Marsh [Mar94]. This model is based on social properties of trust and presents a motivation for integrating some of the aspects of trust taken from sociology and psychology. But having such strong sociological foundations it has been proven that the model is rather complex and cannot be easily implemented in today's electronic communities. Moreover the model puts the emphasis on entities' own experiences, thus neglecting other entities' opinions, so that a network of trust cannot be built collectively.

In this section, those existing approaches that are close to the requirement analysis, presented in Chapter 2, shall be revised. Additionally, this review with respect to the known order of the requirement sequences from the criteria catalogue shall be processed.

As can be seen in the sequence diagram in Figure 3.1, we shall discuss step by step whether the related approaches are appropriate for the prospective solutions (presented in the different scenarios in Chapter 2) and to which extent the given requirements, therein, can be fulfilled. According to the results of this analysis the criteria catalogue will be updated.

3.1 Trust definitions

This section defines relevant terms as used in literature, establishes an analysis to structure the discussion of different trust definitions and dimensions. We will revise and reevaluate the requirements and criteria catalogue we investigated in the previous chapter.

3.1.1 Trust establishment and trust relationships

In most of the approaches related to trust management, one can distinguish between direct and indirect trust. In direct trust the communicating parties verify mutually the authenticity of each others' statements and credentials. This is the most basic form of trust relationships.

When the number of the communicating parties is small, it is possible for each party, for example, to call any of the other parties to verify the validity of the provided credentials. Unfortunately, when the communication involves a lot of principals this would not be an easy task. That is why there is a need to use other mechanisms and, most importantly, to rely on intermediaries to verify the trustworthiness of the communicating parties. These intermediaries are usually called trusted third parties (TTPs).

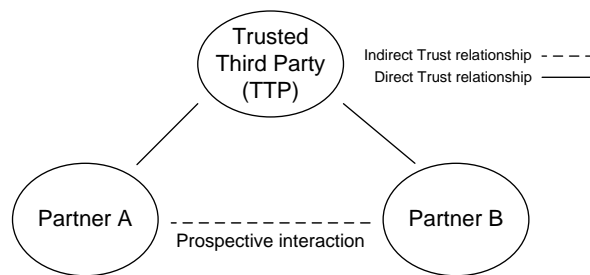


Figure 3.2: Indirect trust relationships with trusted third party (TTP)

3.1.1.1 Trusted third parties in PKI systems

A TTP, by definition, is an independent authority trusted by collaborating parties (either organizations or individuals) when conducting transactions indirectly. As shown in Figure 3.2, a common example for transitive trust relationship using TTP refers to how much implied trust party A gives to Party B when acting on behalf of the trust in TTP.

The most well-known types of TTP is the Certification Authority (CA) in public key systems (PKIs) [Zim94], which serves as an intermediary by verifying public keys and identities and issues certificates using public key cryptography. In PKIs, there are two distinct models that appeal for a trusted third party for building the trust relationship without direct contacts among the communicating parties.

- **The hierarchical trust model:** In this model all the certificates are issued by a third party called Certification, so that if the collaboration partner trusts the CA then he automatically trusts the certificates that CA issues.

This is a simplified form of a hierarchical trust model. In reality there are a number of root CAs from which trust extends. These CAs may issue certificates themselves, or they may issue certificates that are used to issue certificate chains. In doing so, the whole structure is like a trust tree, where the certificate can be verified by tracing backward from its issuer to the issuer's issuer until a directly trusted root CA is found, as it is the case in X.509 standard [X50].

- **Web of Trust:** This model is particularly useful when communicating parties do not use a common CA. Actually the TTP does not have to be a CA and not every party has a certificate from a CA. Instead, the trust relationships, there, can be established indirectly, when the unknown entity provides a certificate that is digitally signed by another entity (TTP) whose digital signature is already known and trusted. Note that this type of verification may go through several intermediaries until a digital signature of a trusted certificate can be found.

A standard implementation of the web of trust model is enforced in the Pretty Good Privacy (PGP) model [Zim94].

3.1.1.2 Shortcomings of the PKI models

In the following, we will discuss the reasons why the hierarchical trust as well as the PGP web of trust models are not suitable for the requirements of the trust management

solution we discussed in Chapter 2.

The hierarchical trust model: As we explained earlier, the basic trust model in a X.509 PKI is a certificate hierarchy consisting of multiple CAs, which in that way enable a chain of trust. Below we delineate some of the limitations of this model with regard to trust management.

- The chain of trust can grow quite long because of the inherited trust that applies up and down the hierarchy. When the partners finally can find a CA that they both trust they can merely use CA certificates to confirm the validity of each other's public key. Apart from key validity, the partners do not have any other information about the trustworthiness of the communicating party in a sense of behavior and reliance.
- Another problem relates to the fact that each party in such a PKI system (e.g. organization) may have its own view on who are the trusted authorities, which may change with time (based on experience). These features as well as the possibility to define its own trust policy (for example to determine which certificates can be accepted) are missing.
- Additionally, these systems have meaningful limits placed upon the use of certificates. There are also limits on the possible parties who could rely on certificates, which makes it impossible to a proper trust management to be realized, because beyond the authenticity of the parties' identities, an evaluation of each party's role and access right in the transactions is required as well.

PGP Web of Trust: For PGP web of trust, which was created primarily for encrypting e-mail messages using public or conventional key cryptography, we delineate the following shortcomings:

- One of the major problems in PGP systems is the lack of fixed or formal certification authority, which considerably increases the uncertainty of the authenticity of any PGP key certificate collected. Further, PGP supports only a <name, key> mapping; it says nothing about the access control (authorization) rights held by the principal.
- PGP has introduced trust levels that correspond to how much the owner of the public-key can be trusted to be an *introducer* to another trustworthy public-key certificate. These trust levels can be one of these (fully, marginal, untrustworthy and unknown). However, the actual meaning of these trust levels is not explicit. Additionally, they can only be used as a rough estimation to how much trust to place in an introducer, because how the user arrives at his opinion about the introducer's trustworthiness is also undefined.
- Absence of security and trust policies; PGP does not explicitly provide any mechanisms for expressing security policies. For compensating the ambiguity of the trust levels, PGP allows its users to tune PGP's *skepticism*. This is done by adjusting two parameters, `COMPLETES_NEEDED` and `MARGINALS_NEEDED` [AR97]. Actually this skepticism mechanism is the closest thing to a policy in PGP [BFL96].

- Another issue relates to the fact that this approach assumes that every public key with the same trust level has exactly the same trustworthiness value, which is clearly insufficient to reflect the highly varying opinions about trustworthiness that a user must put in a public key or introducer.

3.1.1.3 Fulfillment of the requirements?

This discussion attests that the PGP model was never intended to be more than an email encryption software. Furthermore, the trust management requirements like the ones we need to fulfill for our scenarios, must be handled with new mechanisms that are more elaborate than the hierarchical or the PGP model.

Obviously, to consider these new mechanisms, a suitable practical solution requires, among others, a detailed trust levels representation of each trusted introducer. This can be achieved by, for example, assigning more points for the more trustworthy introducer, then define globally how many points are required to fully certify a public-key certificate.

In Table 3.1, we sum up the limitations of these PKI models with regard to the criteria catalogue illustrated in Subsection 2.4.2. However, since these public key infrastructure systems provide identity inspection and assurance for authentication purposes (direct trust) for the trusted third party user, we merely focus on those requirements that are related to indirect trust. Note that the notation used for indicating the fulfillment extent has the following meaning:

- ✓ indicates that the requirement is fulfilled
- indicates that the requirement is not fulfilled

	Indirect Trust	Indirect Trust (by delegation)	Indirect Trust (from past experience)	Indirect Trust (by reputation)	Aggregation
Hierarchical Model	[Trust-Interm]: ✓ [Trust-Level]: – [Trust-Metric]: – [Trust-Context]: – [Trust-Policy]: ✓	[Deleg-Auth]: – [Deleg-TTP]: ✓	[Audit-Info]: – [Audit-Eval]: – [Audit-Stor]: –	[Rep-Value]: – [Rep-Metric]: – [Rep-Context]: – [Rep-Cred]: – [Rep-Recent]: –	[Aggre-Collect]: – [Aggre-Scheme]: –
PGP Web of Trust Model	[Trust-Interm]: ✓ [Trust-Level]: ✓ [Trust-Metric]: ✓ [Trust-Context]: – [Trust-Policy]: –	[Deleg-Auth]: – [Deleg-TTP]: ✓	[Audit-Info]: – [Audit-Eval]: – [Audit-Stor]: –	[Rep-Value]: ✓ [Rep-Metric]: ✓ [Rep-Context]: – [Rep-Cred]: – [Rep-Recent]: –	[Aggre-Collect]: – [Aggre-Scheme]: –

Table 3.1: Fulfillment of the requirements through the PKI Models

3.1.1.4 Trust Metrics

Depending on the type of the relationship, the trust metric reflects different aspects to define the measure of trust or the strength of the prospective relationship. As a measure for trust, the trust relationships might use quantitative metrics, qualitative metrics, or even a combination of these. However, in the literature, there is no real consensus regarding the representation of trust.

Basically the representation of trust can be categorized into two groups: Qualitative and quantitative trust representation. In the following we address some of the possible options for the metrics representing trust in these categories.

Qualitative Trust Metric

Usually the alternative for representing trust values qualitatively is to categorize trust into levels regarding the behavior as well as the performance of the participant. Azzedin et al. in [AM02] use this principle and categorize trust in six levels as: (very low trust level, low trust level, medium trust level, high trust level, very high trust level, extremely high trust level).

A similar scheme and ontology based trust model in the semantic web is given by Golbeck in [ONT06][GHP03], where the levels of trust roughly go from 1 to 9 indicating: (1: Distrusts absolutely, 2: Distrusts highly, 3: Distrusts moderately, 4: Distrusts slightly, 5: Trusts neutrally, 6: Trusts slightly, 7: Trusts moderately, 8: Trusts highly, 9: Trusts absolutely).

However, the usage of these semantics, on the one hand, proves to be very tightly coupled to the application scenario. On the other hand, it does not give an objective view on the trust of the communicating parties. Reasoning about trust in this manner might lead to a lack of accuracy and even to erroneous results, because of the subjective aspect of trust, participants may use different categorization for the same experiences or the other way around.

Quantitative Trust Metric

In order to have a more accurate trust assessment, addressing levels of trust quantitatively is seen as most suitable, although even in doing so, the trust values are not necessarily fixed values associated with the entities, and might not always reflect the partner's real intentions. We argue, however, that they can be applied as trust indicators in specific contexts at a given point of time, more specifically for preventing from eventual risks when engaging in a cooperation.

- *Discrete scale*: The idea behind discrete trust metric is to provide distinct trust levels in which each level specifies a discrete value rather than a range of values for each level. As it is the case in [DZF03], where the level of trust the participants establish to each other can be either boolean or numeric. For example, instead of providing a trust level range from 0 to 1, the array can contain merely two separate levels for 0 and 1. Similarly, Aberer et al. in [AD01] use a sort of *binary* trust indicating that participants are either trustworthy or not.

In Ebay¹ and other eBusiness web providers, the used metric for providing reliable feedback on sellers and buyers of items varies between three values (+1, 0, -1). Another example is enforced in the work of Waguih

¹<http://www.ebay.com>

in [Wag06], which follows the trust level classification of Golbeck [GHP03] and considers a metric consisting of nine grades ranging from absolute distrust to absolute trust within the interval of $[-1, 1]$.

The benefit of this type of scale is that it eliminates any ambiguity about the precise values that are supported. However, a major disadvantage of associating discrete values to trust levels is that they can restrict the expressiveness of the trust assessment approach. For example, if the estimated trust value does not correspond to one known discrete value, obviously, the data range containing this value has to be moved to a position in the array that is ahead or below of the estimated value. We argue that this type of rounding should be handled with care, otherwise it may lead to erroneous results.

- *Continuous scale*: In contrast to discrete scale metric, continuous scale metric has another view on the accuracy aspects. In contrast, it usually represents interval scales and assesses the position of trust levels in a range of values. This type of representation has been conducted in Marsh's PhD thesis [Mar94]. There, he represents trust as a continuous variable using as scale for trust values in the interval of $[-1, +1]$. He states that trust can have threshold values that vary between individuals and situations.

In a similar manner, Sloman in [Slo04] expresses the range of trust levels as integers in $[0, 100]$ where high trust is represented in $[90, 100]$, low trust in $[5, 20]$ and a default initial trust in $[0, 50]$. Negative values represent distrust.

The PageRank algorithm [RD02], used by the Google search engine as a probability distribution, is also based on a trust metric of this kind. It uses the number of links directed to a particular page as votes for that site. This rating, combined with other text processing, is used to score results, where the probability is expressed as a numeric value between 0 and 1. The PageRank algorithm is so effective at rating the relevance of pages, that its results are commonly used as a control for testing the effectiveness of trust metrics.

Another related work that uses such a continuous metric is that of the EigenTrust system [KSGm03]. In the context of peer to peer systems, the EigenTrust system (based on PageRank) effectively computes global trust values for peers, based on their previous behavior. Individuals with poor performance will receive correspondingly low trust ratings. This system was shown to be highly resistant to attack.

Raph Levin's Advogato project² also calculates a global reputation for individuals in the network, but from the perspective of designated seeds (authoritative nodes). Advogato establishes trust between members using a certificate process. Each member can certify his or her trust toward another member in three levels (apprentice, journeyer, and master), and the relationships, there, are based on a metric that composes assertions from members to determine membership within a group. As each member trusts other members, a graph (web) can be built on the basis of who trusts whom. This graph is then traced from the seed member to every other member by the shortest possible route.

Similar to EigenTrust, the Advogato metric is quite attack resistant [GH04], since access to post and edit website information is controlled by these certifications. By identifying individual nodes as *bad* and finding any nodes that certify the *bad* nodes, the

²<http://advogato.org>

metric cuts out an unreliable portion of the network. Calculations are based primarily on the good nodes, so the network as a whole remains secure.

From this discussion, we motivate our choice for using a continuous scale for representing the trust metric. Detailed analysis of the weighting of the trust relationships according to this metric will be provided in Chapter 4.

3.1.2 Circle of Trust (Liberty Alliance Project)

As we stated in Chapter 2 in Subsection 2.1.1, the concept of the CoT originates from the Liberty Alliance Project, which in 2001 was formed by Sun Microsystems together with other major companies. This project is the primary open standards organization for federated identity and identity-based services, and its members represent some of the world's most recognized brand names and service providers.

For this purpose two important sets of standards were adopted and implemented by the Liberty Alliance Project: The Liberty Alliance Project frameworks, and the Security Assertions Markup Language (SAML) specifications. These implementations enable business partners to form a Circle of Trust to conduct network transactions while protecting the individual's identity.

The realization of such a CoT implies that service providers, which offer web-based services to users, join together in order to exchange user authentication information using Liberty web service technologies (Identity Federation Framework (ID-FF) [ID-04a] and Identity Web Services Framework (ID-WSF) [ID-04b]).

Accordingly, the Liberty CoT must contain at least one identity provider, which is responsible for managing and maintaining the users' identities information, so that once a Circle Of Trust is established among the involved service providers, single sign-on is enabled between all these providers.

3.1.2.1 Example of Liberty Alliance Circle of Trust

To illustrate the Liberty CoT, we consider the travel portal as an example of an authentication domain, where the travel portal service forms a partnership, known as a CoT, with a group of hotels, airlines, and car rental agencies displayed on its website. Typically, a travel portal is designed to help the user finding an access to various travel service providers from one Internet location. The user logs into the travel portal and looks for a suitable hotel. When finished making hotel reservations, the user moves to the airline part of the travel portal to look for a suitable airline flight.

Based of the partner agreement and trust relationship with the travel portal in the context of the CoT, the airline website shares the authentication information obtained earlier in the user's online session. Consequently, in a transparent manner to the user, he moves from the hotel reservations website to the airline reservations website without having to reauthenticate there.

Figure 3.3 illustrates the Circle of Trust formed among the travel portal, which acts as the Identity Provider, and each of the related business partners as Service Providers.

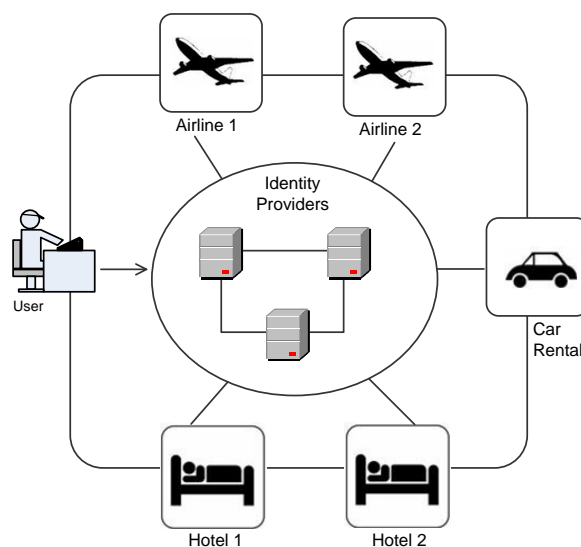


Figure 3.3: Business example of the Liberty Circle of Trust

3.1.2.2 Trust definition in the Liberty Alliance

The concept of trust and trust relationships among the Liberty entities can be broken down into two categories: *Direct Trust* and *Indirect Trust*. However, both of these trust relationships types can be, in turn, derived from two aspects: (i) Authentication relationship and (ii) Business relationship [Lin03]

- **Direct Trust:** Direct Trust is obtained when communicating entities have agreed and determined to trust directly for cryptographic authentication purposes. In this regard, direct trust relationship can be established, when both entities hold each other's keys within their *Trust Anchor Lists (TAL)*³, so that their validity is established without reliance on intermediaries.

Further, two entities may also have a direct trust relationship when they have direct business relationship. More precisely, if an entity requires direct business agreements in order to interoperate with another entity, this entity must be listed in the entity's *Business Anchor List (BAL)*.

- **Indirect Trust:** The same reasoning holds for indirect trust in the Liberty CoT. Indirect trust can be obtained either (i) when communicating entities ascertain the validity of each others' keys based on pre-existing trust established with an intermediary, as represented by a trust anchor, or (ii) through brokered trust, in the case when two entities do not have direct business agreements with each other, but do have agreements with one or more intermediaries so that a business trust path can be constructed between them. The intermediary brokers operate as active entities, and are invoked dynamically from the Business Anchor List⁴ via

³Entities accepting cryptographic authentication of other entities will maintain trust anchor lists, identifying the entities and associated keys that they trust for authentication purposes upon which validations will be based.

⁴The entries in BALs will be added and removed only as a result of explicit administrative action reflecting changes in trust relationships.

		Authentication	
		Direct	Indirect
Business Agreements	Direct	Pairwise/Direct	Pairwise/Indirect
	Indirect	Brokered/Direct	Brokered/Indirect
	None	Community/Direct	Community/Indirect

Figure 3.4: Liberty Trust Models

protocol facilities when new paths are to be established.

3.1.2.3 Liberty CoT models

According to the trust definitions given above about direct and indirect relationships, the Liberty Alliance standards have identified three alternative models in which two entities may be linked to each other via third parties:

Pairwise Trust Model: Pairwise Trust reflects the scenario where two entities have direct business agreements with each other. In addition to the strong trust that is required in a business sense, the cryptographic authentication in these models may be based on pairwise exchange of shared public-key certificates, in conjunction with business/legal agreements.

Brokered Trust Model: Brokered Trust addresses the case where two entities do not have direct business agreements with each other, but do have agreements with one or more intermediaries. These intermediaries may be invoked when federation and/or authentication transactions go beyond multiple administrative domains, and thus, are substantial for constructing a trust path between the requester and the requested entity.

Community Trust Model: Community Trust models presume neither direct nor indirect business agreement paths between communicating entities. Instead, they rely on shared membership and enrolment in a common authentication infrastructure and acceptance of its practices, without reliance on other business agreement paths. Basically, the cryptographic trust establishment infrastructure is used as a basis to enable communication between entities for purposes of federation and authentication.

Public Key Infrastructure (PKI), Kerberos [KER] realms and inter-realm relationships, and PGP web of trust represent examples of the available community trust models.

Table 3.4 distinguishes between these two dimensions of trust. The columns of the table illustrate the types of cryptographic infrastructures applied to support authentication among participants. Proceeding along the horizontal axis, we distinguish between direct authentication (pairwise exchange of cryptographic keys), and indirect authentication (facilitated through the involvement of trusted intermediaries).

Here, the rows illustrate the types of business agreements established between participants as a basis to support transactions. Proceeding along the vertical axis, we distinguish between direct agreements, indirect agreements, and the absence of business agreements linking participants, which generally typifies the Trust Community Model.

3.1.2.4 Shortcomings and limitations

The majority of the shortcomings and limitations of the Liberty CoT Framework with regard to the issues addressed in the previous scenarios, discussed in Subsection 2.2, relate to the following:

- Most of the Liberty CoT models extend the focus on authentication of partners and rely on digital signatures with the enforcement of business agreements for enabling transitive trust building and users' identities federation. The combination of these mechanisms provides confidence in the source of the partner, which is very important, but trust in this sense ignores the credibility issue. That is because confirming the source of the requester does not have any explicit implication about the quality of the prospective relationship.
- Further, the Liberty CoT models enable interorganizational web single sign-on. However, none of these models can be applied to services which are not yet or cannot be fully web enabled, e.g., e-mail and file storage services [HR05].
- In the pairwise trust models as well as in other brokered trust models, relationship and business trust between all interoperating participants are exclusively governed by signed business agreements (either directly or indirectly). Obviously, the strong trust establishment via business agreements is not technically extendable, which results in forming closed communities. That is, a new entity may not interact within such a community without first entering into a business agreement with the existing participants and being added to the Business Anchor List.
- Regarding the security and privacy aspects, beside the communication security that is considered in the standards (mainly in the public key infrastructure based solution), the Liberty models enforce, in a static manner, few measures for protecting the privacy of the users' data when disclosing and transferring information across members' domains.

3.1.2.5 Fulfillment of the requirements?

Based on this analysis, we see that the Liberty solution provides a means for the members of establishing transitive trust relationships on the basis of the authentication mechanisms as well as the business agreements that relate them. We deduce, however, that the notion of trust, in the context of our requirements analysis, regards merely the delegation of rights as well as the organizational aspects.

Table 3.2 recalls the criteria catalogue and gives an overview over the extent to which the requirements are fulfilled in the discussed areas (mainly for the requirements with regard to direct trust, indirect trust by delegation, organizational aspects, and privacy management, because indirect trust from past experiences or by reputation is not supported).

Direct Trust	Indirect Trust (by delegation)	Organizational requirements	Privacy Management
[SEC-AAA]: ✓ [SEC-Pol.]: ✓	[Deleg-Auth]: ✓ [Deleg-TTP]: ✓	[ORG-TLA]: ✓ [ORG-Time]: ✓ [ORG-Sim.]: – [ORG-Cost]: – [ORG-Integr]: ✓ [ORG-Impact]: ✓	[Priv-Coll]: ✓ [Priv-Use]: ✓

Table 3.2: Fulfillment of the requirements via the Liberty CoT Models

3.2 Indirect trust dimensions

In this section, we review related works in the context of indirect trust. Here we refer to the techniques that build trust from indirect collaborations, for example by using observations from third parties as well as other behavior indicators, so that if two entities are not directly connected, an indirect trust inference uses the paths that connect them (see discussion on indirect trust in Subsection 2.1.2.1).

3.2.1 Indirect trust by delegation

Well-known trust models that have introduced delegation or inferring trust include Pretty Good Privacy models as well as the simple public key infrastructure (SPKI) [EFL⁺99][E11]. SPKI, which was motivated by the inadequacy of public-key infrastructures based on global name hierarchies, such as X.509 [ITU93], is used for authentication and authorization but only includes a simple notion of delegation. While in PGP, as we discussed in Subsection 3.1.1.1, an entity is trusted when one or more trusted entities declare it as trustworthy. Both of these schemes suffer from key distribution problems and do not deal with flexible or scalable access control.

One of the known approaches that addresses the issue of building trust by delegation is the one described by Kagal et al. [KFJ01]. This approach, basically designed over a policy-based framework that extends SPKI and role-based access control, deals with trust issues particularly in pervasive computing environments, where central authentication strategies are inadequate for the increased flexibility that these environments require. This is due to the fact that these systems usually have a lack central access control mechanisms and their users are often mobile and not all can be predetermined.

This approach allows delegation chains in which users are able to delegate their rights to other users they trust. Once users are given certain rights, they are responsible for the actions of the users to whom they subsequently delegate those rights and privileges, thus, building the so called delegation chains.

Enabling dynamic delegation of rights aims at extending the security infrastructure with a distributed trust management solution, which involved developing a security policy,

assigning credentials to entities, verifying that the credentials fulfill the policy, delegating trust to third parties, and reasoning about users' access rights.

Shortcomings

Although both requirements for trust by delegation ([Deleg-Auth] and [Deleg-TTP]) are regarded in these approaches, reasoning about trust only by delegation presents two major challenges:

- Reasoning about trust from delegation in such open models presents challenges at many levels. By definition, users usually can access a service if they have the right to do so or if an authorized entity has delegated that right to them. Further, they can delegate all rights that they have the permission to delegate. Although the delegated rights can likewise be revoked, there are, however, no ways to control how the access to the given service has been performed.

That is, if any user along the delegation chain fails to meet the requirements associated with a delegated right, the chain will be broken. However neither the behavior nor the reason of such failure can be archived for eventual future interactions.

- The second challenge relates to the fact, that in absence of authorized entities who can make delegations and revocations in the form of signed assertions, there will be no other alternatives to reason about the trustworthiness of unknown entities, and subsequently provide appropriate access rights.

3.2.2 Indirect trust from past experience

Assessing the trustworthiness of individuals from the aspects of past experiences is a perceptual process and usually can be seen from different facets. Reasoning about trust in this way is particularly relevant in several federated environments such as in Grid Computing, where resource providers belong to distinct administrative domains, each manages its policies and resources with a high degree of autonomy, and where the entities, exploiting the Grid infrastructure, typically have incomplete information about each other.

For the Grid computational resources that are typically executed by applications on behalf of *unknown* grid users, several related works for preserving the integrity of the resources exist. Most of these works orient themselves toward distributed auditing mechanisms and fine-grained monitoring of the actions performed on the resources.

In the context of auditing, Cederquist and al. [CCD⁺07] introduced a framework for modelling ownership of data and controlling compliance to the data policies in academic collaborative environment. In this framework, a formal audit procedure by which users may be audited and asked to justify that an action was in compliance with a policy, was defined.

The main characteristic of this approach is that the compliance of users to policies is checked a-posteriori. On the one hand this yields a more flexible system for the users, but on the other hand it requires that users take responsibility for their actions. This is handled through the following assumptions:

1. In order to make it possible for auditors to observe critical actions, there must be

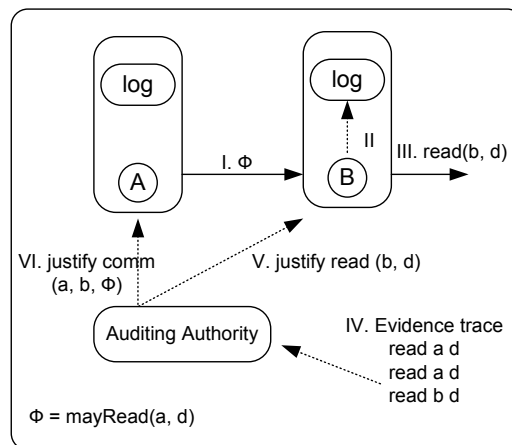


Figure 3.5: Sample deployment depicting actions such as the logging and the interaction with an auditor [CCD⁺07]

a sufficiently comprehensive audit trail, which cannot be forged or bypassed, and which should contain the relevant details about the actions and the identity of the users executing them.

2. For holding accountability of the actions performed by users, it is required that data about the users will be stored after joining the system.

Based on these assumptions, as illustrated in Figure 5.13 for a scenario of collaborative document editing, a sample run of this framework works as follows: In the first step (I), agent A provides a policy ϕ to agent B that will be recorded in the log of agent B (II). Next, in step (III) agent B reads document D .

At this point in time, no assumption can be made on whether the document D may be stored or not. At a later point, in step (IV) the auditing authority, which is checking access to sensitive files, finds the access of B and requests B to justify this access (V). In response to this request, B shows that the access was allowed according to the policy ϕ which was provided by A . Based on this statement, the authority auditor, initially unaware of A 's involvement, can now (VI) audit A for having provided the policy ϕ to B .

In this study, building trust from logging and auditing information mainly addressed issues related to compliance with the predefined access policies. Other similar approaches considered further information and attributes, which might be helpful for deducing trust from the audit data.

Chopra and Wallace [CW03] in their framework for trust in electronic environments consider the trust in entities according to several factors and processes that contribute to trust. One of the most relevant factors is the *competence* of the partners collaborations while interacting between each other. Competence, in this regard, implies that the trustee possesses the knowledge, expertise and ability to fulfil the needs of the trustor.

Competence in Grid environments in [Pap08] is regarded by the attributes of (i) *Correctness* indicating that the participant delivers the proper outputs or payments, and (ii) *availability and accessibility* indicating that the participant is available and running as

is expected.

Doney et al. introduced in [DC97] the notion of *Belief* as a related attribute for reasoning about trust. It represents the degree to which information provided by the trustee can be believed directly in relation with past experiences and prior collaborations.

However, other attributes reflecting how trust can be derived from the audit data of communicating parties is that of *credibility*. According to Peters et al. in [PCM97] perceptions of trust and credibility are dependent on three factors: perceptions of knowledge and expertise; perceptions of openness and honesty; and perceptions of concern and care.

Due to the fact that usually experts are perceived as being more trustworthy than the others, in Grid environments, the assessment of expertness is related, to some extent, to breadth of knowledge or depth of knowledge, offered by the specific participants as a solution for a specific problem. An optimal solution to a difficult problem is most convincing with respect to the capabilities of the parties involved in the collaboration [Pap08].

Shortcomings

The actual approaches for reasoning about trust from past logged experiences represent a number of limitations for both aspects of collaboration trust as well as content quality trust (see Subsection 2.2.2.1). In the following we delineate those limitations that relate to the applicability of these systems in interorganizational collaborative scenarios and to the quality of the trust assessment:

- Most of these systems are not intended for use in interorganizational scenarios, basically because only behaviors of internal and known entities can be recorded. Although not countervailed, some of these issues are addressed in previously discussed approaches, at least to some extent, by, for example, suggesting that audit trust be evaluated not just through direct relationships, but also at the account of intermediaries. This concept is enforced in the approach of Cederquist and al. [CCD⁺07] in collaborative document editing environment, but there, again, the audit system can only audit how known entities were involved in providing policies and rights to unknown entities, so that the behavior of these unknown entities remains untraced.
- In the context of interorganizational collaborations, these limitations include the lack of unified representation of the attributes and the indicators for trust that can be monitored during and after interactions (for example the attributes of correctness, credibility, etc). This, obviously, involves investigation on scheme and metric representation. Furthermore, the aggregation aspects for extracting a general value of trust represent an important challenge when assessing trust according to the given attributes.
- In Grid systems, these trust attributes can generally be represented as Quality of Service parameters (as we discussed in Subsection 2.2.3.3). Based on these parameters, the parties may gain confidence that an interaction party will offer the desired QoS and consequently behave as expected. We believe that in some settings these assumptions are not realistic, and thus, there is a need of a flexible and easy way to audit collaborations according to these QoS parameters, considering different roles (consumer or provider), and thus assessing trust accordingly.

3.2.3 Indirect trust by reputation

As we discussed in Chapter 2, beside the principles of *trust by delegation* and *trust from past experiences*, trust can be appraised by reputation as well. In contrast to the other two dimensions, reputation, in this context, is more regarded as a social notion of trust.

In several related works reputation has been defined as a measure that can be derived from direct or indirect knowledge on earlier interactions of agents, and which can be used to assess the level of trust an entity puts into another entity. Thus, reputation-based trust management is one specific form of trust management.

Several recent works investigate reputation as a substantial dimension for managing trust in open and distributed systems [YS02]. The model of Zacharia for electronic commerce interaction in [ZMM99] defined two complementary reputation mechanisms that rely on collaborative ratings (direct experiences) and personalized evaluation of the various ratings assigned to each user (recommendations from other parties), where the reputation values are defined as subjective properties assigned particularly by each individual.

Similarly in electronic commerce, Schillo et al. [SFR00] presented a formalization, where agents can observe the behavior of others from a directed graph (nodes representing the agents and edges the information on the most recent reputation rating given by the agents) and thus collect information for establishing an initial trust model. In order to adapt quickly to a new or rapidly changing environment, they enable agents to make use of observations from other agents.

However in both models, the rating, which represents the reputation value does not relate to the context of the interaction. Additionally, neither the credibility of the rating nor the reliability of the agent for providing ratings are investigated, for example, to detect inappropriate recommendations.

Abdul-Rahman and Hailes in [ARH00] addressed the issue of credibility, by developing a model that allows agents to decide which other agents' opinions they trust more and, thus, allows agents to progressively tune their understanding of another agent's subjective recommendations. Still this subjectivity is quite limited because of the lack of mechanisms for mapping context description with the given ratings.

3.2.3.1 Reputation management in Peer-to-Peer systems

A standard application area for reputation management is that of Peer-to-Peer networks in which all peers cooperate with each other to share resources and perform functions in a decentralized manner. In such environments, usually, there are no centralized control authorities and all peers can be both consumers and providers of resources [XL02].

In that sense, Aberer and al. [AD01] consider the semantic of reputation essentially as an assessment of the probability that an agent will cheat. The global trust model considered in this study is based on binary trust, i.e. an agent is either trustworthy or not, because agents perform transactions and each transaction can be either performed correctly or not. If an agent cheats within a transaction, it will be labeled as untrustworthy for future transactions.

3.2.3.2 Reputation management in eCommerce

Over the last years, mainly due to the arrival of new possibilities for doing business electronically, many researchers started to recognize the importance of trust management in electronic communities.

An important practical example of reputation management in eCommerce is eBay⁵, the largest online auction site. In eBay, registered users can offer items for sale by auction. Each user can be identified by a pseudonym he may choose himself. After each transaction buyers and sellers have the opportunity to rate each other and the overall reputation of a participant is computed as the sum of these ratings over the last six months. Of course, a main characteristic with this approach is that everything is completely centralized at the data management level.

Similarly, visitors at Amazon⁶ often look for customer reviews (the reviews may be negative or positive) before deciding to buy new books.

3.2.3.3 Reputation management in the Semantic Web

On the semantic web, trust and reputation can also be expressed using dedicated ontology, which usually provides methodologies for describing entities and the trust relationships between them. Golbeck in her PhD Thesis [Gol05][GHP03] investigated algorithms for social networks making use of these ratings. In her study, she developed metrics as measurement scale to infer relationships among entities in the semantic web and to extract trust information about them.

Moreover, in this study a trust assessment solution has been created to extend the Friend of a Friend (FOAF) project⁷ (which comprises data and files about persons in social networks) in order to allow persons to create ratings for one another. This solution is realized in a project⁸ that provides tools and support for producing trust data that can be linked to an aggregator.

The PageRank algorithm [RD02], used by the Google search engine, is also a trust metric of sorts. It uses the number of links coming into a particular page as votes for that site. This rating, combined with other text processing, is used to score the results, mainly for rating the objectiveness and the practicality of the published pages.

The PageRank algorithm is so effective at rating the relevance of pages, that its results are commonly used as a control for testing the effectiveness of trust metrics.

3.2.3.4 Shortcomings

As we discussed earlier, the common objective of most of the presented works is to assess the trustworthiness represented by the reputation of the peers by collecting some feedback parameter values reflecting, for example, satisfaction, complaint as well as context of the transaction. However, in the following we will depict some of the limitations that these solutions entail:

⁵<http://www.ebay.com>

⁶<http://www.amazon.com>

⁷<http://www.foaf-project.org>

⁸<http://trust.mindswap.org/>

- In most of the online reputation systems, it is notable that the users whose reputation is mainly negative are seldom found [KMW00], due to the fact that users can easily get a new pseudonym for the respective service, and thus recover from a negative rating.
- In the same context, another problem that faces the reputation systems is the lack of mechanisms for detecting and avoiding unfairly high ratings and unfairly low ratings, since users might give one another or even themselves unfairly high ratings, usually by creating a second pseudonym for the same service.

Although, there does not exist any kind of discrete mechanism for dealing with false information provided. Some reputation systems like CNET⁹, EPINIONS¹⁰ and ALLEXPERTS¹¹ compute reputations based on the feedbacks of experts and reviewers, assuming thus, the presence of experts and qualified reviewers in the environments when the reputation feedbacks are needed.

According to [Gal04], these problems can be handled to some extent by hiding the mapping of pseudonyms to users and assigning random pseudonyms for each login, which makes providing unfair ratings more difficult. Besides, cluster algorithms may be used to filter out unfairly high ratings [Del00]. However, both approaches cannot be applied in all electronic commerce scenarios, because it is not always possible nor desirable to hide the users' true identities just by assigning random pseudonyms.

- Moreover, in the majority of these works all peers are treated equally as opinion makers, which render them too simple (in terms of their trust rating values and the way they are aggregated) for applications in open FEs. In contrast to these works, our interorganizational scenarios have much more complex roles distributions in collaboration efforts and much more requirements on building trust than by reputation from third parties' statements and experiences.
- As we will discuss in Subsection 3.2.5 regarding the fulfillment of the requirements from Chapter 2, setting up reputation management systems of feedback and ratings is undoubtedly helpful to create environments where participants feel safer when collaborating together, but obviously, reasoning about trust by reputation alone is not adequate in every collaborative environment.

3.2.4 Indirect trust aggregation

All the approaches mentioned before have in common that they strive to extract trust from different dimensions, from identity verification to interpersonal social interactions. However, there are very few works on aggregation of trust, at least in the form required in the scenarios, we presented in Chapter 2.

A known model presented by Jennings et al. in [HJS04] and [HJS06], presents a trust and reputation model called FIRE that integrates a number of information sources to produce a comprehensive assessment of an agent's likely performance. To this end, the

⁹<http://www.cnet.com>

¹⁰<http://www.epinions.com>

¹¹<http://www.allexperts.com>

FIRE model incorporates four types of reputation information (interaction trust, role-based trust, witness reputation, and certified reputation) to provide a trust metric in virtually all circumstances.

However, the alternative source information, which the agent is able to combine in this model basically evolves around reputation and rating information, which proves to have certain limitations. For example, if agent *A* has not interacted with *B* before (e.g. agent *A* has just joined the environment) and no other agents did, there will be no other information that help agent *A* to assess agent *B*'s trustworthiness.

In such situations, we argue that in case agent *B* can present certified information about its past performance to *A* (for example in the form of references from other agents who have interacted with it (delegation), or there are mechanisms that audit the way past experiences have been performed, agent *A* will then be able to make some assessment of its trustworthiness.

Based on that, developing a general framework for structuring and aggregating the trust information from different aspects and dimensions in federated environments and shared information spaces is the primary goal of this thesis.

3.2.5 Fulfillment of the requirements?

On the basis of the discussion about the shortcomings of the revised related works in each of the indirect trust dimensions, in Figure 3.6 we recall the relationships among the corresponding requirements from the criteria catalogue, and accordingly summarize the extent to which they can be fulfilled therein. Note that the notation used for indicating the fulfillment extent means the following:

- ✓ indicates that the requirement is fulfilled
- ~ indicates that the requirement is marginally fulfilled
- – indicates that the requirement is not fulfilled

As can be seen in this illustration, while reasoning about trust by delegation has been widely investigated in previous research areas (for example in [KFJ01] as well as in several public key systems), a lot of effort is needed in the field of evaluating trust from past experiences and by reputation. Especially for adaptability issues with regard to interorganizational collaborations (the CoT) and, more importantly, with regards to aggregation aspects.

3.3 Interorganizational access control mechanisms

A wide range of access control models has been proposed in the past years to address the security needs for managing and assigning access to services and resources. These models are categorized as either mandatory access control (MAC) models or discretionary access control (DAC) models [DAC93], depending on how the policies are specified [BFL96]. A mandatory security model is designed to control the flow of sensitive information according to the users' security clearance.

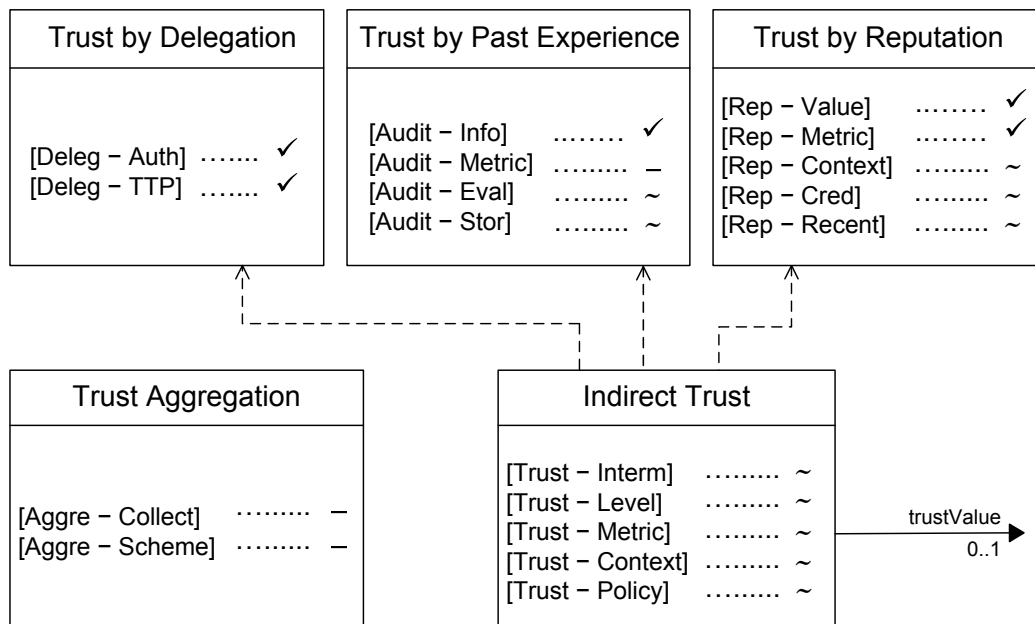


Figure 3.6: Fulfillment of the requirements with regard to indirect trust in federated environments scenarios.

The DAC model is characterized by its flexibility in controlling data access based on the users' identities. Moreover, it allows users to grant authorization to other users. The access control model used in most operating systems and database systems follows this model.

3.3.1 Intraorganizational access control models

A newer alternative approach to MAC and DAC models is that of Role-Based Access Control (RBAC) [FBK99][SFK], which extends those models by assigning permissions to perform certain operations based on specific roles. For example, in contrast to the access control lists (ACLs) used in traditional discretionary access control systems, RBAC system assigns permissions to specific operations with relationship to the user's role in the organization, rather than to low level data objects (e.g. file system). Accordingly, since users are not assigned permissions directly, but only acquire them through their membership (role), the management of users' rights simplifies common operations such as adding a user, or changing a user's department by simply assigning the appropriate roles to the users.

The Task-Based Access Control [TSY94] builds on the RBAC model for providing high-level semantics for security specifications. Abstractions such as *role* and *task* can be introduced to bridge the semantic gap between enterprise-level policies and low-level security rules.

3.3.2 Extension tentatives to interorganizational scenarios

These access control standards are widely accepted as a best practice within a single system or application and have successfully been applied to intra-organizational scenarios (different operating systems, data bases and LDAP directories effectively implement some form of RBAC). However, they have later been extended for interorganizational and federation scenarios, in order to allow the delegation of administration on the one hand and privileges on the other hand.

However, for this purpose, the concept of RBAC is based on the identity, local role, or capabilities of the resource requestor, which proves to be inappropriately course-grained access control because this solution does not scale. An alternative solution that provides a more appropriate course-grained access control is realized with the attribute based access control (ABAC), whose goal is to overcome these granularity and scalability problems by providing a means for each holder of authority to determine and specify its own judgments that can be combined naturally to make appropriate authorization decisions.

Another approach that aims at extending the RBAC model beyond one organization's boundaries is the one described by Kagal et al. [KFJ01], where initially access decisions are based on the roles that individual users have through their membership in the organization. In Subsection 3.2.1, we discussed that the extension of the trust management framework allows delegation chains to be established, in which users are able to delegate their rights to other users they trust. Concretely, this concept involves developing security policies, assigning credentials to users, verifying that the credentials fulfill the policies, delegating trust to third parties and reasoning about users' access rights after the delegation has been carried out.

3.3.3 Shortcomings and fulfillment of the requirements

Unfortunately, these access control models do to prove to be a good starting point for interorganizational collaborations in FEs and for the inclusion of external entities. In the following, we shall depict some of their limitations in this regard:

- It is obvious that privileges may only be delegated to those principals which are already known in the organization in particular or in the federation in general. This means that a digital identity that has been created by one of the involved organizations must be assigned to the user a priori, which causes the very same timeliness, cost, and complexity problems we strive to avoid.
- Within the same argumentation, these models by themselves are not sufficient to define and enforce interorganizational level security policies. This is because they were developed in the context of a single organization for controlling users' access to resources, and do not possess enough constructs to represent interorganizational security polices and constraints.

However, we strongly believe that trust-related concepts and constructs such as trust agreements, delegation, trust by reputation and past experiences should be integrated within those existing access control models for modelling and managing privileges on

Interorganizational access control requirements

[Access-Auth]	✓
[Access-Policy]	–
[Access-Storage]	–

Table 3.3: Fulfillment of requirements for interorganizational scenarios

the one hand and enabling resource owners to retain authority on their resources on the other hand.

Based on this discussion, in Table 3.3, we revise the fulfillment of the interorganizational requirements (see Subsection 2.2.1.5 for more details on these requirements) through the presented access control models.

3.4 Policy control

General models of trust frequently cannot be separated from works in security and policy representations, because trust and policy management are related to each other, with dependent concepts that may have different representations. In the context of trust, policies can by definition serve to express, for example, when, what and how trust in a participant can be determined, as well as precisely how much trust is required in a principal in order to be allowed to perform a certain action on a given resource.

According to our definitions, this type of policies are investigated in the trust computation and evaluation algorithm as discussed in the requirement [Trust-Policy] on page 43.

However, in this subsection, we focus merely on policies that relate to the privacy aspects, for example privacy concerns when managing, storing and distributing the trust data (for example the trust levels, the reputation values, etc.) among the members of the CoT.

3.4.1 Privacy management

Winslett et al. in [YW03] have considered the tight relationship between privacy and trust establishment. For the purpose of achieving an automated trust negotiation, they investigated access control policies, which can be associated with sensitive credentials to control the circumstances under which those credentials can be disclosed. An additional related work in the context of privacy control, where constraints of personal user's data release can be managed, has been realized in [Hom07] through the extension of the XACML [Edi05] Attribute Release Policies.

However, this study focused mainly on the privacy of disclosing identity credentials in open systems, while our requirements in the context of privacy address especially issues, for example, on how far recommended trust information can be transmitted to other parties, how this can be modified, how it can be the base of a trust decision, etc.

Another approach, based on these principles is the approach of the Trust-

Privacy Management requirements		Risk management requirements	
[Priv-Collect]	~	[Risk-Level]	~
[Priv-Use]	~	[Risk-Metric]	~
		[Risk-Rule]	-
		[Risk-Update]	-

Table 3.4: Fulfillment of requirements for privacy management and risk management

Builder [WYS⁺02]. In this approach, trust can be established between strangers by gradually disclosing credentials until a certain degree of confidence is achieved among them. We argue that these concepts may be a good starting point for exploiting the privacy of the trust data in the CoT, and therefore, will be part of the objectives of this thesis.

3.4.2 Risk management

As we discussed in Subsection 2.2.3.5.2, risk management is also categorized under the shape of policy management. This section gives a short overview of the related work in risk management in relationship with trust management and its influence on our approach:

Mayer et al. [MRD05] complement the integration of security aspects in requirements engineering by adapting and integrating risk analysis in the iterative cycle of information system development. While this proposal strives to identify the existence of risk that affects the assets of IT systems, it does not assess the level of risk quantitatively.

The same limitation is encountered in the approach of Lee et al. [SLA05], which investigates the interactions between various models within their framework, and considers the relationships between security requirements and risk assessment. This framework investigates the mappings that exist between the security requirements enforced by the standard of the Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP) [JES00] and the elements of risk assessment to carry out a justifiable risk assessment process. However, this risk assessment process is exclusively bound to the DITSCAP ontological characteristics and lacks from establishing common-understanding risk metrics.

In the area of risk quantification, the SECURE project [Jen02] worked on a framework that considers the trust in a principal as well as the risk for granting its request. The policy language used in SECURE uses a simple grammar, which is not sufficiently expressive to encode risk metrics.

Similarly, [BZ02] and [YS02] consider policy-driven decision making by evaluating trust and the impact of countermeasures; these two approaches make use of thresholding in their policy language for comparing the trust-values with a certain level of reliability. However, these threshold values are statically determined and fail to consider any run-time evaluation of trust and risk values, which obviously limits their flexibility.

In table 3.4, we summarize the limitations of these contributions in light of the requirements we discussed in the field of privacy and risk management.

3.5 Organizational Trust

In this section we shall consider trust from the organizational point of view. Most of the Internet-based collaborative applications involve interorganizational interactions. In order to ensure the protection of the assets of all parties involved in these collaborations, interactions must be regulated by a contract, as it is the case within traditional business interactions.

3.5.1 Defining Trust by law

The term of eContract has emerged in eCommerce scenarios in the last years. Basically, eContract can be regarded as a digital facilitation or automation of a contract in a cross-organizational business process. A basic e-contracting architecture for Business-to-Business (B2B) scenarios was proposed in [MB95]. It includes key elements like a contract repository, contract notary, contract monitor, and contract enforcer. These key elements interact with each other in the following way:

- The contract repository stores standard contract templates, so that when two organizations choose a contract template and agree upon the content, the contract notary stores the contract.
- The compliance with contract terms is verified and ensured by the contract monitor and the contract enforcer, which monitor, regulate and control all business interactions that have been agreed upon according to the given contract.

This approach proposes a general framework for business contracts by addressing two main issues: i) The inadequate representation of the semantics of business activities and ii) the lack of a sound legal support for electronic interactions. Other related work in the area of e-contracting includes the EU-funded COSMOS project [GBW⁺98] and the CrossFlow ESPRIT project [KGV00][Hof99].

In the same context, [CCT03] presents a methodology for the engineering of e-contracts enforcement from a high-level document-view down to the implementation layer based on this architecture, using a supply-chain example. As a result, e-contracts can be seamlessly defined and enforced. Conceptual models of various layers are given in the Unified Modelling Language (UML).

We note that the meaning of trust in these approaches is very limited to the agreements upon the negotiated contracts. However, Brainov et al. in [BS02] deepened a bit the aspect of trust in these application areas and analyzed the impact of trust on market efficiency, and especially on multiagent negotiations. They defined the advance payment as a solution to the problem of distrust and as a screening device for separating trustworthy agents from untrustworthy agents. In doing so, they proved that every advanced payment is not only individually rational, but it also maximizes the amount of trades.

The Service Level Agreement (SLA) from IBM [LKDK02] is another research effort that studies agreements with respect to qualities of services (QoS), such as throughput and downtime. The SLA, in this case, specifies the QoS requirements.

3.5.2 Discussion

Unlike these approaches, our research focus is on the specification and enforcement of trust agreements with respect to the entities' behaviors as well as the interorganizational security policies and constraints. We envision that our work will eventually be integrated within existing circles of trust platforms (CoT is usually composed of a set of organizations), so that the access control decisions can be based on the estimated trustworthiness of an unknown entity.

However, we argue that the principles of certified e-contracts are helpful to make a clear separation between the global policy and the local policy in the CoT. All agreement specifications can be enforced as rules and configuration data into the global policies to manage collaborations among the members of the CoT, whereas, local policies support local autonomy, which is an important requirement in designing a trust-based security model for supporting collaborative environments.

3.6 Content quality trust

In Chapter 2, in both scenarios in Subsections 2.2.2 and 2.2.3, we have discussed the fact that trust can be based on the quality of the content of the resources as well as the services being shared in the FEs – for example, the consumer trusts the service to provide the necessary functionality as well as quality. However, many different usages of the aspect quality can be found in the literature.

A close discussion of trust from the aspects of quality is given by Friedman et al. [FPKH00], where characteristics of trust in online interactions are outlined. One of the key points presented there is that simply performing a task is definitely not the same as providing good service of high quality.

Several examples motivate the use of QoS, which include parameters like accuracy, precision and performance, so that the entity can be regarded as trustworthy based on those QoS requirements whose fulfillment is supposed to be ensured. Maximilien et al. in [MS04a], [MS04b] and [Max05] investigated the possibility of dynamic service selection in web services via an agent framework coupled with a QoS ontology. In this approach, participants can collaborate to determine the non-functional attributes regarding quality and trustworthiness of each other service. In the given ontology, QoS attributes are the key features for dynamically selecting the services that best meet the user's needs. These attributes can either be objective, representing, for example, reliability or availability aspects or can be subjective by focusing on users' experiences.

However, these approaches have a lack metrics which may help the consumers to reason about how providers meet the QoS. Moreover, it cannot be shown how the predefined ontology scale with the increasing number of services in the environment and how the service providers scale with the changing number of QoS properties. Furthermore, they evaluate only service providers from the consumer's point of view and do not offer any possibility for the providers to evaluate the trustworthiness of other providers and consumers.

Another application field where QoS can be applied for assessing trust is within Grid computing environments. According to Papalilo [Pap08], the overall assessment of the QoS can be performed by Grid participants themselves, since they are the ones able

to evaluate the efficiency of the services offered and the grade of fulfillment of their requirements.

In doing so, the collected information provides the participant with the means for assessing the various quality characteristics of its collaboration parties. Consequently, the behavior of the collaborating parties can be rated accordingly, so that only those parties that prove to have fulfilled the requirements, described in the QoS properties, are regarded as trustworthy.

We conclude that in both approaches the requirement [Content-Quality] (see page 64 and page 71) with respect to QoS is considered to some extent, while the other requirements regarding the storage and run-time evaluation of this new type of trust data are not addressed.

An additional important case study, where the trustworthiness of the partners is strongly down to the quality of their provided content, is the one of Wikipedia source information. In the following, we shall take a closer look on the problems of trust in Wikipedia.

3.6.1 Wikipedia Case Study

The intrinsic characteristics of Wikipedia¹² make the utilization of trust solutions challenging, due to the fast changing nature of the online distributed articles, which can be viewed as large enough to provide challenges of scale and trust; and due to the increasing need of trust for storing much rich provenance information in comparison to typical collaborative information repositories.

Although the Wikipedia feature of the speed at which the articles can be updated (the most visited and edited articles reach an average editing rate of 50 modifications per day, while articles related to recent news can reach the number of hundreds of modifications [DBWS06]) is designated to be as one of its strongest features, this dynamic aspect affects considerably the validity of the standard trust techniques used for evaluating the articles' origins and content quality.

One emerging objective of online collaborative information repositories is to enable different groups of users to collaborate in a distributed manner to create and maintain a repository of shared content. The notion of open editing has become popular along with the notion of the community-built and freely available online encyclopaedia Wikipedia, which in its simplest form allows users to freely create and edit web pages about different topics and subjects¹³.

On the one hand, the size and the diversity of Wikipedia content repositories are aspects that make them an interesting collaborative content management. Having more than 900,000 registered authors¹⁴ and with more than two million of entries in English on a wide variety of topics – with versions in dozens of other languages, they have become a valuable resource and many users cite it as a credible information source.

However, on the other hand, the perplexing aspect of Wikipedia is that it uses the collective knowledge of its public contributors and editors. Since no charges are included in this kind of distributed editing, when misusers or other history-revisionists strike, those same contributors are tasked with keeping content of articles' content in the *neu-*

¹²<http://www.wikipedia.org>

¹³<http://wiki.org/wiki.cgi?WhatIsWiki>

¹⁴<http://en.wikipedia.org/wiki/Special:Statistics>

tral point of view. Note also that the quality of Wikipedia varies from entry to entry, and that some helpful quality indices already exist in order to refer to the information as being not totally untrustworthy or biased.

Content Trust in Wikipedia

While recent studies (e.g. [Gil05]) try to prove that the science articles in Wikipedia are generally trustworthy, there have been some reports of claimed inaccuracies and even erroneous information appearing in Wikipedia. For example, there was a widely reported incident where a journalist and a former official in the Kennedy administration, stated that Wikipedia contained an inaccurate biography article about him in 2005 [SEI]. This incident raised questions about the reliability of Wikipedia and other online shared information repositories that lack accountability and consideration about issues of trustworthiness of content sources.

After this incident, Wikipedia took some steps to prevent editing faulty and inaccurate information, such as excluding unregistered users from creating new pages [Hel05], and imposing more control over the existing liberal editing policies of anonymous authors.

However, it is additionally important to note that Wikipedia as an open (or mostly unrestricted) editing environment is quite different from some other static online communities that have addressed trust, as we discussed the case of online eLearning community in the IntegraTUM scenario in Subsection 2.2.1. Because other social networks may be viewed as focusing on interactions between users while generating interactions' feedbacks as a growing content but not typically generating changing content. For example, a feedback review on the transaction on eBay is typically created once the interaction has ended and then remains unchanged but can be commented by the other side.

By contrast, the content of collaborative information repositories like Wikipedia may be quite dynamic as it may be continually reviewed, shared, and updated by many different users. Accordingly, the trust formulation and requirements for rapidly changing repositories thus may be quite different from monotonically growing repositories.

Content Trust by reputation in Wikipedia

Based on this discussion, it is obvious, that effort on trust management in collaborative information repositories such as Wikipedia cannot be efficiently evaluated by reputation, in a sense that the authors are rated according to the quality of their articles or documents. This is because, reputation systems in this context are user-driven, which would require users with rich knowledge about the published topics, in order to be able to rate each other's contribution's quality.

Adler and al. [AdA07] addressed the issue of trust in Wikipedia differently. They developed a content-driven reputation system for Wikipedia authors, where they do not use the concept of user-to-user comments or ratings. However, authors can gain reputation when the edits they perform to Wikipedia articles are preserved by subsequent authors, and in the opposite case, they lose reputation when their edits are rolled back or undone in short order.

However, association rules or simple revision parameters (such as the number of revisions) are not very useful in computing and tracking trustworthiness of articles that

Content Quality Trust Requirements			
[Content-Quality]	~	[Store-Complex]	~
[Content-Rep]	✓	[Store-Monitor]	~
		[Store-Conflict]	-

Table 3.5: Fulfillment of the requirements for Content Quality Trust

are under constant change. For example, a featured article could become untrustworthy if it has been changed despite the fact that the number of revisions is monotonically increasing.

Zeng and al. [ZAD⁺06] explored ways for utilizing the revision history of an article to assess the trustworthiness of the article, and thus, of the owner of the article. The trustworthiness of the revised version depends on the trustworthiness of the previous version, the author of the last revision, as well as the amount of text involved in the last revision. Next to the metric that is basically chosen rather arbitrarily (with four different trust levels representing administrators, registered users, unknown users and blocked users), an author is trustworthy, for example, when he is likely to make a large amount of changes to a very untrustworthy article. In this approach, the trustworthiness of the author may be based on his interest in a given article as well.

However, it is obvious, that these assumptions might be subject to debate and cannot be applied in interorganizational scenarios, where, for example, undesired contributions from malicious or careless users cannot be easily undone. Therefore, we assume that we can profit from these approaches for the fulfillment of the requirements [Content-Rep] and [Storage-Complex] in line with FE and CoT, but still, the associated metric must be investigated and adjusted.

3.6.2 Shortcomings and fulfillment of the requirements

Based on the discussions on related works in the field of Content Quality Trust, Table 3.5 shows the grade of fulfillment of the requirements falling in this category.

Though the [Content-Quality] and [Content-Rep] are the two most important requirements in determining the trustworthiness of an entity through the quality of its resource or service, other requirements, such as the complexity of the storage, run-time evaluation of the stored trust information and eventual conflicts, for example, when the trustworthiness is also evaluated from other aspects, are just as important. It is also important to note that these issues did not receive enough attention in the current efforts on trust management in federated environments.

3.7 Prototypes – Solutions for automated trust assessment

Several existing works for automated trust assessment, which are based on the fundamentals we discussed in the previous subsections, have influenced the development of

our Trust-Based Access Control (TBAC) solution, the architecture as well as the prototype implementation. We discuss them below:

3.7.1 PolicyMaker and KeyNote

PolicyMaker [BFL96][BFIK99] was developed by the AT&T Labs as a unified approach for trust management systems whose main goals relate to privacy, authenticity and anonymity as security requirements. It specifies and interprets security policies, credentials and relationships that allow direct authorization of security-critical actions. Moreover, it expresses security credentials and policies without requiring the application to manage a mapping between personal identity and authority.

The access control model of PolicyMaker is very similar to the one of Kagal [KFJ01], because it also enables the expression of conditions that specify cases under which an individual or an authority can be trusted. More precisely, it extends the authentication of users' identities by specifying what a public key is authorized to do (evaluates whether a proposed action is consistent with a local policy). In this regard, policies are trust assertions made by the local system and are unconditionally trusted by the system. Credentials are signed trust assertions offered by other entities whose signatures must be verified before their usage.

KeyNote [BFIA99], as a successor of PolicyMaker, was developed to enforce the way security rules and digital credentials can be used for security policy enforcement in a distributed system. As an extension of PolicyMaker, KeyNote accepts as input a set of local policy assertions, a collection of credential assertions and a collection of attributes (action environment) that describe a proposed trusted action associated with a set of public-keys in simpler programming language (C-like programming language). Applying assertion predicates to the environment makes verifying the consistency of actions with a local policy possible. The result of the KeyNote evaluation process is an application-defined string, basically indicating *authorized* or *unauthorized*.

3.7.2 Trust Policy Language (TPL)

A similar approach to PolicyMaker and KeyNote has been conducted by IBM, which developed the Trust Policy Language (TPL) [HMM⁺00] for defining trust policies for web services. Within this framework, it can be attested that the trustworthiness of an entity involved in an e-business transaction, which is subject of verification, can be resolved by using certificates. Usually, those certificates can be issued by various participants, vouching for a specific participant in a particular role (buyer, seller or both).

Further, the *Trust Establishment Module*, which encloses all the policies and the rules that map a web service requestor to one predefined role or permission according to the provided certificate, validates the client's certificate and defines what a role is permitted to do.

Following the same argumentation as for PolicyMaker and KeyNote systems, beyond validating a principal's certificate and mapping the certificate owner to a specific role, the TPL trust model does not consider any aspect for tracking or representing principals' behavior nor aspects for dynamic update of these.

3.7.3 REFEREE Trust Management Model

Rule-controlled Environment For Evaluation of Rules and Everything Else (REFEREE) [CFL⁺97], as a joint effort by researchers from AT&T Labs and W3C¹⁵, aims at creating a general-purpose trust management system for Web applications. By providing both a general policy-evaluation mechanism (for web clients and servers) and a language for specifying trust policies, REFEREE places all trust decisions under explicit policy control.

This model is based on PolicyMaker and considers trust problems for PICS labels¹⁶¹⁷ as the stereotypical web credential and uses the same theoretical framework as PolicyMaker to interpret trust policies and administer trust protocols. All these are represented as software modules.

Similar to PolicyMaker and KeyNote, REFEREE is a recommendation-based query engine, which by design can be integrated into a host application. It evaluates requests and returns, as a justification for the answer, a statement-list, which can be represented in three possible outcome values: *true*, *false* or *unknown*.

3.7.4 Standards for the World Wide Web

The basic idea behind establishing trust in the World Wide Web evolves from basic mechanisms and primitives for providing secure messaging between parties. As stated earlier, in order to secure a communication between two parties, the two parties must exchange security credentials (either directly or indirectly). However, each party needs to determine if they can *trust* the asserted credentials of the other party.

In the following, we delineate some standards for trust in the World Wide Web and discuss how trust has been defined therein:

Security Assertion Markup Language (SAML)

SAML [SAM03] provides a protocol that is able to transfer information about entities between various cooperating domains without the need for those domains to lose the ownership over that information. Due to the fact that the exchanged information can be represented as assertions related to a subject, its authentication or authorization information, it investigated a standardized way to securely exchange this type of information between trading organizations regardless of the security systems or platforms in use.

The aspect of SAML has been widely used in e-business transactions across company boundaries by means of the trust assertions.

¹⁵<http://www.w3c.org>

¹⁶<http://www.w3.org/PICS/>

¹⁷Platform for Content Selection (PICS) provides rules that together form a kind of filter between the web documents and their viewers based on policies. It was developed by the World Wide Web Consortium to protect primarily children from pornography on the Internet. PICS offers some rating that determine the appropriateness of a target internet page.

WS-Trust

Recently, IBM, Microsoft, Verisign, and RSA have collaborated and proposed new specifications that regard trust in web services, namely the *Web Service Trust Language (WS-Trust)* [[WST04](#)].

Using the Web Service Description Language (WSDL), the Web Services Trust Language (WS-Trust) defines messages and operations for the issuance, exchange and validation of security tokens in order to enable applications to construct trusted SOAP¹⁸ message exchanges. We conclude that this trust is represented through the exchange and brokering of security tokens.

Although the specification includes the description of a general message model for trust establishment through security token exchange, this model does not specify how collaborating organizations come to an agreement and establish interorganizational security policies, and how the agreement enables the collaboration between these organizations. Moreover, the management of trust policies among these domains is not addressed.

XML Key Management Specification (XKMS)

The XML Encryption WG has developed an XML-based cryptographic technology to preserve confidentiality of data elements that are represented as XML documents [[XML](#)]. The XML Key Management Specification (XKMS) [[XKM](#)] is merely an XML-based PKI service that enable distribution and management of the keys that are necessary for ensuring end-to-end communication security. Thus, digitally signing and encrypting XML documents is actually the only fashion for trust establishment.

3.7.5 Shortcomings of these automated trust assessment systems

In comparison with the previously given definitions about trust, trust within these approaches is tightly bound to the classical security measures, so that, here, it is either present or absent. It is defined as the output of the identity and authorization verification process, thus, after credentials and their claimed associations are verified. That is, beside the possibility of rights' delegation, the notion of trust in the context of the CoT is quite limited. Here we discuss further issues in more detail:

- The term of distributed trust management, coined by those approaches is mainly related to the fact that public keys can be bound to access control without authentication. Due to this fact, these approaches may be regarded as a query engine that answers questions about access rights to a given policy rather than a trust management system with the dimensions described in the previous chapter.
- Moreover, although these systems are seen as powerful analytical tools [[IPS02](#)], the non-programmers who are likely to develop policies for collaborating in heterogeneous environments with different backgrounds may have difficulty expressing policies in these systems, which may lead to a major deficiency, especially for the objective of dynamic evaluation of trust in federated environments.

¹⁸<http://www.w3.org/TR/soap/>

Due to the fact that delegation of rights as well as trust and authorization policies are supported, we conclude that the presented systems for automated trust assessment fulfill the requirement to the same extent as the hierarchical models presented in Subsection 3.1.1.3 as well as the delegation models presented in Subsection 3.2.1.

3.8 Analysis and conclusions

In this chapter we discussed a wide number of related research contributions in the field of trust management and access control. Naturally we focused on environments where the communicating parties belong to different domains, and thus, we again demonstrated that trust in most of these contributions can be broken down into two main categories:

1. *Collaboration Trust (indirect Trust)*; these approaches focus on trust aspects in particular with regards to the authentication and authorization of users, so that every participant in the environment is known through its identity. In this regard, trust on the identity reflects the confidence on the declared identity of the participant.
2. *Content Quality Trust*; Beside the confidence on the identity of the requester, during the collaboration, trust depends on the quality of the performance of the participant as well. Content quality trust, in this sense, reflects the confidence in the quality of the collaborating partner's services or provided resources.

3.8.1 Discussions

As we discussed in several sections, in the category of collaboration trust, trust can be established either (i) by delegation, (ii) from past experiences or (iii) by reputation. However, the approaches we revised in this context are relatively limited to a single dimension. For example, for those approaches that relate to PKI infrastructures, they define specific situations for building trust, e.g. X.509 specifies trust only in context of creating reliable certificates, PGP for key introduction etc.

While there are very few works on trust from past experiences, contributions for trust by reputation make use of the term *trusted*, but focus explicitly on assembling ratings and critics from other partners, ignoring thus other important aspects, such as the context of the interaction as well as other critical aspects like the credibility of the rating.

The second category (Content Quality Trust) has also been subject of some studies. However, apart from describing some parameters for defining the quality of the collaboration, little agreement on what trust from content quality really is, how the content quality can be characterized, and how it can be mapped to trust, has been achieved.

Aggregation of trust

Another common limitation with the majority of the proposed approaches is that they are used to identify a static form of trust. Trust is mostly evaluated only at the start of a collaboration considering a unique source of information (a unique dimension) either direct experiences or information from third parties either about the identity of the requester or the quality of its past collaborations.

Using only these information sources, however, imposes several additional concerns that deal with the poor direct experiences or with the subjectivism of third parties' opinions. Therefore, the absence of *aggregation* mechanisms for putting together different source information about trust and for providing overall trust values represent one of the most pertinent objectives of this dissertation.

Inter-Organizational Access Control

With regard to interorganizational access control and trust management policies, the majority of the related approaches represent trust as the probability of a binary event, that is, the probability that a partner will cooperate or defect.

However, by modeling a partner's possible actions simply as cooperation or defection, several factors that may have effects on the assessment of trust are ignored, such as privacy and risk management. For example, it is not clear how the parties can determine whether to release a certain credential or data in spite of the possible presence of risk (e.g., privacy violation) and what negotiation strategies are possible when automatic access decision cannot be preformed.

Change Management

In addition, the intentions and the behavior of the participants are often subject to changes and evolutions. We noticed that the need to monitor trust relationships to determine whether the criteria on which they are based still apply is not regarded in the previously discussed approaches.

3.8.2 Update of the criteria catalogue

Based on this discussion, we conclude that many of the proposed approaches differ significantly in their definitions and computational methods for trust. Although many of the above models claim to handle various aspects of trust, they have failed to consider the need of aggregating trust from different information types, such as aggregation trust from the three dimensions (delegation, past experiences and reputation) or aggregating those aspects with that of content quality for the management of access control in federated environments.

Based on the previous discussions, the focus of our study can be summarized in the following tasks:

- Aggregation of trust from different source information.
- Access control model based on trust and policy control to define and enforce interorganizational access and collaboration policies and constraints on top of the existing systems. This model should also enable participants to incorporate their own preferences in the decision-making process.
- Runtime evaluation and change management; since trust is dynamic, a notion of learning and adaptation is therefore required in order to be able to adapt to changing conditions of the environment in which the trust decision was made.

To conclude this chapter, in Table 3.7 we recall our criteria catalogue from Chapter 2 and sum up the extent of the fulfillment of the requirements through the presented approaches.

Note that in this table, the same notation, which we used for expressing the extent of the requirements' fulfillment, is used here to delimit the contribution of our work, with regard to the bulk of requirements as follows:

- ✓ indicates that the given requirement was addressed in the so far discussed approaches and is regarded as fulfilled. This implies that our approach shall use integrally the best suitable approach that fulfills this requirement for our scenarios. Details as well as references to the usage of each approach will be given in Chapter 4.
- ~ indicates that the requirement was marginally fulfilled; however, there are still some inadequacies that need to be addressed. This implies that our approach shall build on the approach that fulfill the requirement and adjust its insufficiencies.
- indicates that the requirement is addressed in none of the previously discussed approaches. This implies that our approach shall investigate a completely new mechanism for filling this gap from scratch.

In the next chapter, the trust model that aims at fulfilling the remaining requirements, in the form of the objectives cited above, will be presented.

<p align="center">Direct Trust Requirements</p> <p>[SEC – AAA] ✓ [SEC – Policy] ✓</p>		<p align="center">Indirect Trust Requirements</p> <p>[Trust – Intern] ~ [Trust – Level] ~ [Trust – Metric] ~ [Trust – Context] - [Trust – Policy] ~</p>	
<p align="center">Trust by Delegation</p> <p>[Deleg – Auth] ✓ [Deleg – TTP] ✓</p>	<p align="center">Trust by past experience</p> <p>[Audit – Info] ✓ [Audit – Metric] - [Audit – Eval] ~ [Audit – Stor] ~</p>	<p align="center">Trust by reputation</p> <p>[Rep – Value] ✓ [Rep – Metric] ✓ [Rep – Context] ~ [Rep – Cred] ~ [Rep – Recent] ~</p>	
<p align="center">Trust Aggregation</p> <p>[Aggre – Collect] - [Aggre–Scheme] -</p>			
<p align="center">Interorganizational Access Control Requirements</p> <p>[Access – Auth] ✓ [Access – Policy] ~ [Access – Stor] ~</p>		<p align="center">Technical Realization Requirements</p> <p>[Tech – Integrity] ✓ [Tech – Protocol] ✓ [Tech – Storage] -</p>	
<p align="center">Organizational Requirements</p> <p>[ORG – TLA] ✓ [ORG – Cost] - [ORG – Time] ✓ [ORG – Integr] ✓ [ORG – Simple] - [ORG – Impact] ✓</p>			
<p align="center">Policy Control Requirements</p>			
<p align="center">Privacy Management</p> <p>[Priv – Collect] ~ [Priv – Use] ~</p>		<p align="center">Risk Management</p> <p>[Risk – Level] ~ [Risk – Metric] ~ [Risk – Rule] -</p>	
<p align="center">Change Management Requirements</p> <p>[Sec – Update] ✓ [Risk – Update] - [Trust – Update] ~ [Notify] - [Rep – Update] ~</p>			
<p align="center">Content Quality Trust</p> <p>[Content – Quality] ~ [Store – Complex] ~ [Content – Rep] ✓ [Store – Monitor] ~ [Store – Conflict] -</p>			

Figure 3.7: Fulfillment of the requirements in light of the criteria catalogue

Chapter 4

Trust Process Model

"Trust is generated over the course of an extended history... Contrary to belief, it cannot be designed. It has to be built – little by little."

Brigitte Jordan

Contents

4.1	Conception of the trust process model	127
4.2	Phase 1: Initialization	128
4.2.1	Modeling Trust	128
4.2.2	Trust Assessment	136
4.2.3	Content Quality Trust and QoS Trust	159
4.2.4	Aggregation between the three dimensions of collaboration trust	162
4.3	Phase 2: Storage and management	165
4.3.1	Organizational models	165
4.3.2	Data structures	167
4.3.3	Risk managements aspects	169
4.4	Phase 3: Validation	170
4.4.1	Establishment of Trust Agreements	171
4.4.2	Policy Control	173
4.5	Phase 4: Evolution	177
4.5.1	Monitoring	179
4.5.2	Assessment and evaluation of the monitoring information	179
4.6	Phase 5: Auditing and Change Management	180
4.7	Evaluation and conclusion	180

In Chapter 3, we have dealt with a wide variety of related works in the field of trust management, and that the concerns about opening up internal operations and release of private data are less critical between long-established partners, thereby creating static circles of trust.

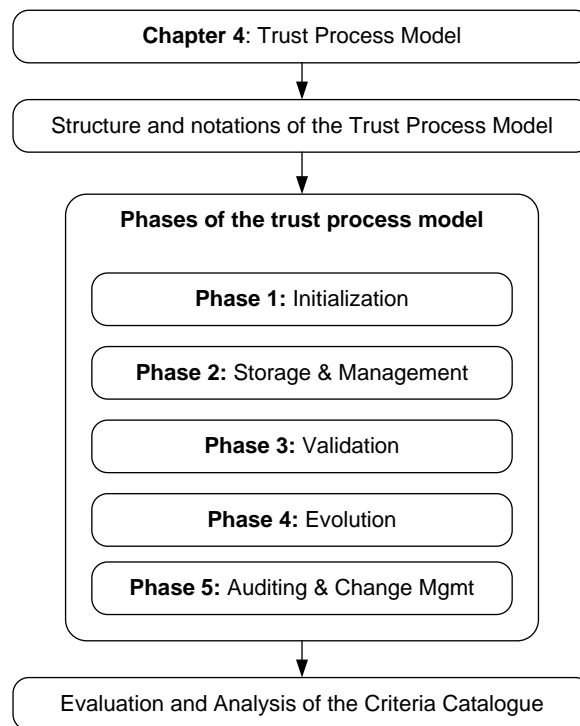


Figure 4.1: Sequence structure for Chapter 4

However, it becomes a major issue in the case of new or evolving partnerships, or collaborations that are meant to exist briefly or to be created dynamically, for example just only for a single transaction. Therefore, trust in dynamic or virtual CoTs (e.g. virtual organizations in Grid Computing), is an important issue as it can have the greatest impact on ensuring the durability of these environments.

On the basis of the discussions we provided on these related works as well as their limitations to fulfill the requirements for our TBAC solution, in this chapter, we present the trust process model that aims at filling these gaps, especially in the following areas:

- Mechanisms for assessment and aggregation of trust levels that should take into account a variety of sources of trust information in order to have a more precise trust measure (by cross correlating several perspectives) and to cope with the situation when some of the sources may not be available.
- Management of access control based on trust and risk levels, which prove to be very important for providing essential ways to allow the parties to defend the private data they hold against inappropriate access or use by others. Given the *no central authority* nature of federated environments, parties will typically be unwilling to rely solely on a single centralized trust and/or reputation service.
- Run-time evaluation and update of the trust and risk information about the shared resources and services, including, for example, the change of the trust in the potential partners; trust in the warrantors and authorities (if any), trust from the quality aspects, etc.

Following the sequence given in Figure 4.1, first, we provide formal terminologies and basic notations for the elements and attributes we shall use in our model for representing the trust relationships with regard to the CoT. Subsequently, we present our trust process model as a collection of activities designed within different phases to manage trust and trust relationships within and across the CoT.

This study implies a specific ordering of the activities in each phase. In the last section of this chapter, we shall, then, discuss the results and show the strong emphasis on how the criteria catalogue can be optimized by means of concrete examples, from the initialization of the trust relationship until the access control decision and the archiving activity.

4.1 Conception of the trust process model

In Chapter 2, we have demonstrated by means of several scenarios, that successful realization of the CoT vision of a broadly applicable and adopted framework for federated and virtualized environments requires substantial support for managing the trust relationships among the involved parties.

The deployment of these trust relationships as well as the management of their lifecycle, obviously, demand a number of components and computational techniques that need to be set together in a sequence of a process model. In this thesis, we designate such a process model as a **Trust Process Model**.

As we introduced in Chapter 1, in Subsection 1.3.1, our trust process model is mainly composed of five phases. In the following we revise shortly what each phase is responsible for:

- 1st **Phase:** Initialization; or *instantiation* represent the phase where the trust relationships among all participants either inside or outside the CoT, are initialized, computed by the different search and computation methods and finally aggregated. The weights of these relationships are typically represented by trust levels. The state of failed search (unknown trust level) is also regarded within this phase.
- 2nd **Phase:** Storage and Management; this phase contains activities that are more related to issues for distributing, storing, and accessing the resulted trust information among the involved entities. This includes primarily models for data structures and schemes, unified description of the shared resources as well as access control rules on the trust information under the consideration of the privacy policies in the CoT.
- 3rd **Phase:** Validation; the information resulting from the previous two phases help, in this phase, to make better management choices especially for those entities that do not possess static credentials in the CoT. To achieve this, centralized as well as decentralized authorization policies may not be neglected.
- 4th **Phase:** Evolution; Obviously, information about trust levels, access rights as well as resource description are not static but change, for example, based on delegations and revocations. In this regard, an auditing evaluation method of the previous phases will be performed both quantitatively and qualitatively.

5th **Phase:** Auditing and Change Management; this phase is very close to the *Evolution* phase. That is, based on the evaluation results, it considers the change management process in order to update the trust-related information and ensure a runtime evaluation of this information.

In the following sections, we shall study each phase separately and show how concretely the objectives therein are realized.

4.2 Phase 1: Initialization

As mentioned above, this phase represents the initialization step of our trust process model. The main task, there, focuses primarily on dynamic assessment of trust for not necessarily known principals, in such a way that it can be conveyed, for example, that trust has in some specific manner a relative aspect, like principal *A* may trust principal *B* with respect to relation *X*. Moreover, the results of this phase shall be used in the loop of the process model as a trust knowledge base for future interactions and similar inquiries.

Accordingly, the objectives of this phase can be broken down into three main tasks:

- **Modeling Trust:** It identifies the aspects pertaining to the trust assessment procedure in order to establish unified notations and schemes for representing the principals in the CoT, the trust-related attributes and metrics, query dimensions, and all the related aspects that can be needed in the trust quantification process.
- **Assessment of trust:** On the basis of this formal representation, different types of algorithms for assessing trust from: (i) past experiences, (ii) by delegation and (iii) by reputation will be investigated and analyzed.
- **Aggregation of trust:** Finally, this step handles the verification of the different results (if any), aggregates them and generates the appropriate trust level of the partner regarding several functional as well as non functional parameters.

4.2.1 Modeling Trust

The initial situation for modeling trust is when beginning an interaction with a new participant, the trust management model initially requires principals to gather some knowledge about their counterparts' characteristics, for example, reputation or any other indicators about their *behavior*. To be able to automate such a task, a unified and standard data model is therefore required.

To illustrate the different dimensions and terminologies that are needed for quantifying trust, we review the formal class definitions given in Chapter 2 in Subsection 2.1.2.3. We follow the same argumentation, given previously, for representing the prospective trust relationship as a template from which the trust related aspect and dimensions can be instantiated and derived.

As we see in Figure 4.2, the key features of this model are identified as follows: Any trust relationship between two principals is associated with a value (also denoted as a

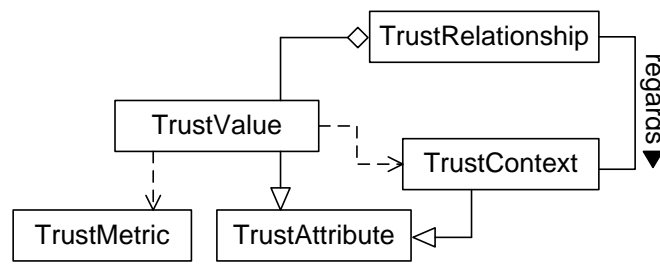


Figure 4.2: Basic relationships of trust definitions

trust level), which basically represents the expectations or the strength of trust in the communicating principal. Moreover, these values can only be quantified according to a given metric (either quantitative or qualitative).

From the realization point of view, in this thesis, the trust values can be represented as attributes that are created along with the identity profile of the principal. These attributes are meant, on the one hand to serve as an evidence of past experiences as expectations for the future. On the other hand, they reflect further information about the trust relationship. One important aspect that the trust attributes reflect is the context of the interaction, indicating precisely how and under which circumstances the trustworthiness has been evaluated.

In the following we shall detail the used notations within each of these definitions:

4.2.1.1 Principals

Entities are represented as a set of principals $\{P_1, \dots, P_n\}$ who participate in the CoT, or have direct and indirect collaboration relationships to the members in the CoT.

Thus, principals relevant to our approach can be divided into several groups: organization members of the CoT (*member*), CoT-external organizations that are known by CoT members, external organizations whose identity can be verified (for example by a certificate issued by a *Certificate Authority CA*), and unknown organizations. These principals can assume different roles.

4.2.1.2 Trust Metric

For presenting and linking the principals with trust relationships, obviously, a metric that allow the creation and encoding of the trust values from the different trust dimensions is needed.

In Chapter 3 in Subsection 3.1.1.4, we brought forward the argument that the usage of qualitative metrics for encoding trust does not give an objective view on the trust in a participant and may lead to a loss of sensitivity and accuracy. We also discussed that, for the objective to have a more accurate trust assessment, numeric values are seen as most suitable.

Therefore, in this work, we shall use a quantitative numeric trust metric. Note that precise representation of computed trust values requires a continuous scale, while fa-

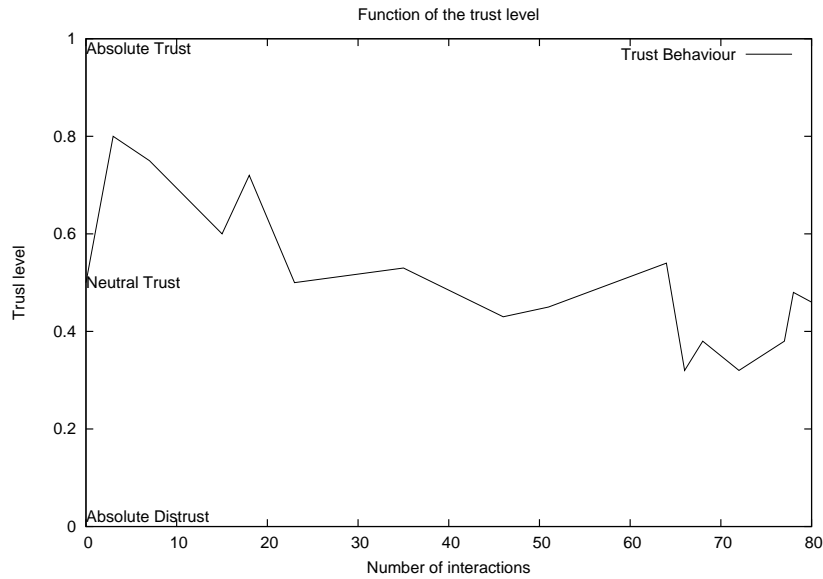


Figure 4.3: Representation of an exemplary trust behavior graph

miliarity between principals may use special discrete values to represent established –rather than calculated– trust.

Accordingly, we define trust as having values in the continuous range $T_l \in [0, 1]$, where 1 indicates absolute trust and 0 indicates absolute distrust. By using an intermediate value 0.5, which indicates a neutral trust, principals with trust values greater than 0.5 are regarded as trustworthy to different levels. Equivalently, principals with trust values lower than 0.5 are regarded as untrustworthy to different levels as well. They may also be undefined in cases where numerical values cannot be found; this is usually the case when there are no principals with direct relationship to the unknown entity. We assign a value of -1 to indicate an unknown trust level.

As we will discuss in the course of the next sections, the trust values are subject to change after each interaction that takes place within a collaboration. Therefore, we conclude that the computed trust level $T_l(t)$ at any point in time t within the cooperation lifecycle T can be always computed in function of the previous state and the change from (t) to $(t - 1)$:

$$T_l(t) = T_l(t - 1) \pm \Delta T_l \quad (4.1)$$

According to this rule the trust level T_l can vary arbitrarily in the interval $[0, 1]$ in function of the amount of interactions as shown in Figure 4.3. In the following subsection we define an update function that deals with the potential change ΔT_l , such as increasing or decreasing the trust values with regard to the number of interactions.

Update function

The update function, presented in Equation 4.2, is composed of three functions:

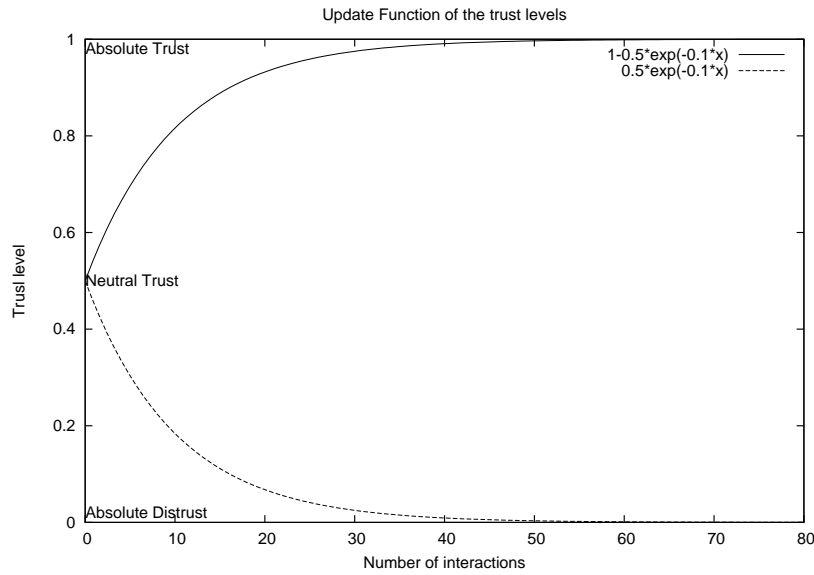


Figure 4.4: Update function of the trust values

$$\Delta T_t = \begin{pmatrix} 1 - \frac{1}{2}e^{-\alpha(\sum interaction(\chi))} & \text{if } 0.5 < \chi < 1 \\ 0 & \text{if } \chi = 0.5 \\ \frac{1}{2}e^{-\alpha(\sum interaction(\chi))} & \text{if } 0 < \chi < 0.5 \end{pmatrix} \quad (4.2)$$

- $1 - \frac{1}{2}e^{-\alpha(\sum interaction(\chi))}$ as a function of the number of interactions, realizes the incrementing curve of the trust values that represent trustworthy principals (above the 0.5 axis). As shown in Figure 4.4, the curve increases (viewed left-to-right), but never touches the axis of 1 (indicates absolute trust) in such a way that the value of trust gets more and more close to it by requiring a considerable amount of interactions. This function, on the one hand, facilitates a rapid incrementing of the trust level from the Neutral Trust Level to a higher trust level, but on the other hand, it ensures that sufficient positive interactions have to be performed in order to reach the Absolute Trust Level.
- The function $\frac{1}{2}e^{-\alpha(\sum interaction(\chi))}$ is the inverse function of the previous one. The same principle is defined for all the trust values that represent untrustworthy principals. Similarly, the curve is climbing fast for trust values getting lower than 0.5, while it climbs slowly when approaching the 0 axis for representing Absolute Distrust Level.
- 0 is a static function indicating that no changes are required by the Neutral Trust Level, which naturally may improve or degrade according to the two functions described here.

Where $interaction(\chi)$ represents the set of interactions with a rating level χ . Note that depending on the application scenario, χ can be represented in discrete values: $\chi \in \{0, 0.5, 1\}$ as well as in subjective metrics such as $\chi \in \{negative, neutral, positive\}$. We shall detail the application of this update function with some application examples in both approaches *trust from past experiences* and *trust by reputation* in Subsections 4.2.2.3 and in 4.2.2.4.

Detailed Seller Ratings (last 12 months) ?		
Criteria	Average rating	Number of ratings
Item as described	★★★★★	93
Communication	★★★★★	89
Shipping time	★★★★★	93
Shipping and handling charges	★★★★☆	93

Figure 4.5: Criteria of the feedback ratings in eBay

The parameter α represents the convergence factor of the exponential function curve. Figure 4.4 shows that for $\alpha = 0.1$ the trust level converges to 1, respectively to 0 after approximately 50 interactions. Note that this parameter can be freely adjusted in function of the requirements of the CoT scenario.

4.2.1.3 Trust context

As we have illustrated in Figure 4.2, the notion of context can be considered as input for the decision-making process. Context of trust refers to situational details characterizing the nature of the trust relationship between entities, as well as the environment surrounding them. These details, in turn, can be considered to make a difference with regard to whether to trust or not.

Based on that, we consider trust between two principals to be established for a certain *trust scenario*, in analogy to trust relations between people: Trusting someone to cooperate with you on a task may be different from trusting their *opinion* about a third party. Therefore, we differentiate between trust scenarios $S : \{S_1, S_2, \dots, S_n\}$ by situational interactions (depending on the collaborative environments, these scenarios may be specified, for example, for the shared services, or for certain actions on the shared resources, etc), and we consider queries with respect to third parties to be just another scenario, for this purpose.

In the setting of the IntegraTUM scenario in Subsection 2.2.1, the transmission of account information from the identity provider (TUM), the access to the content provider (LMU) and (LRZ) services as well as the peering agreement between these providers regarding network traffic can all be formulated as trust scenarios.

Another example is that of the eBay rating system. There, all ratings that an eBay user receives are associated to a given set of contexts, defined as criteria (see Figure 4.5). These criteria (such as the *Shipping time*, *Communication*, etc) provide more details about the member's performance as a seller, where five stars is the highest rating, and one star is the lowest. At the end, the ratings are summed up into a *Feedback Rating Number*, which will be attached to each member's ID.

4.2.1.4 Matrix representation

Based on the definitions given above, in relationship with *trust metric* and *trust context*, trust values can be assigned or computed. They are employed to quantify the level of

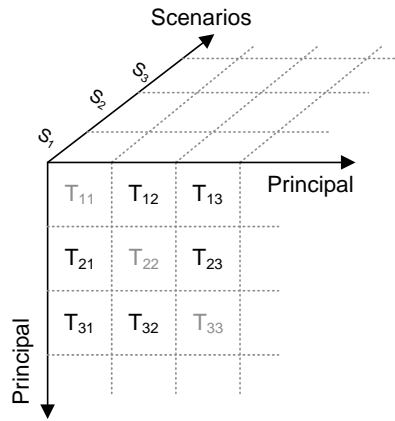


Figure 4.6: interorganizational trust with respect to scenarios

trust placed by one principal in her relationship with another principal. We use the value Set $T : \{T_{l_{1,1}}, T_{l_{1,2}}, \dots, T_{l_{n,n}}\}$ to represent the computed trust relationships.

Additionally, we have determined that trust relationships are formulated with regard to a pair of principals in the context of a scenario. Hence, a three-dimensional structure (Figure 4.6) is necessary to represent the trust relationships between the members of a CoT as well as their relationships to other principals that may be located outside the CoT. The height and width of the cube represent the members, while the depth of the structure represents the scenarios.

$$M(S_k) = \begin{pmatrix} - & T_{l_{1,2}} & T_{l_{1,3}} & \dots & T_{l_{1,n}} \\ T_{l_{2,1}} & - & T_{l_{2,3}} & \dots & T_{l_{2,n}} \\ \vdots & & \ddots & & \vdots \\ T_{n,1} & & \dots & & - \end{pmatrix} \quad (4.3)$$

Note that an example of this matrix with a single scenario S_k can be represented in a two-dimensional matrix with the entries $T_{l_{ij}} \in [0, 1]$ as shown in Equation 4.3.

4.2.1.5 Graph representation of the CoT

We defined the circles of trust as federated environments that facilitate business cooperations among a set of organizations while ensuring that security and privacy requirements are met. This definition implies the establishment of static trust relationships among the members of the CoT.

Figure 4.7 illustrates the trust relationships (edges) between organizations (circular nodes). We differentiate between the relationships within the CoT, those crossing the border of the CoT and those located outside the CoT. From a FIM perspective, every organization can assume the role of identity provider, service provider, or both.

On the basis of these input parameters, we start by mapping the CoT to a graph G . We consider a member in the CoT as a node of the graph G and a path between two

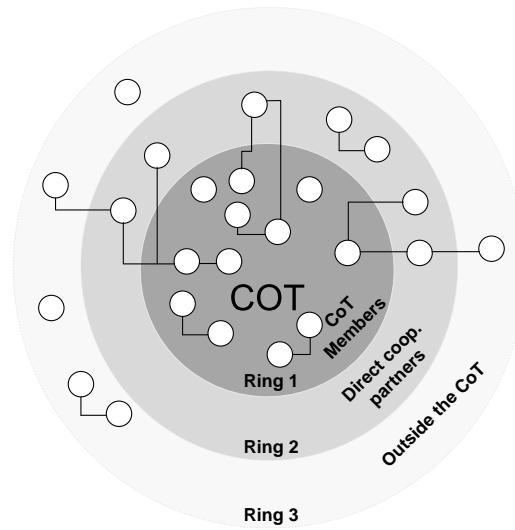


Figure 4.7: Business and trust relationships

members of the CoT as an edge of it (e.g. the path from node 0 to node j can be represented in this form $T_{l_j,k} \leftarrow \dots \leftarrow T_{l_{2,1}} \leftarrow T_{l_{1,0}}$). The graph G can be further defined as follow:

For the objective of creating a finite set of trust relationships, a graph $G = (V, E)$ is defined as having a finite set V of vertices (also denoted as a set of nodes representing the principals) and a finite set $E \subseteq V \times V$ of edges (represent the trust relationships between them). The transitive closure $G^* = (V^*, E^*)$ of a graph $G = (V, E)$ is defined to have $V^* = V$ and to have an edge (u, v) in E^* if and only if there is a path from u to v in G .

Accordingly, the investigated path between the pairs of apparently distant principals is consequently an alternating sequence of nodes and edges, beginning at a node P_1 and ending at a node P_j , and which does not visit any node more than once. In the next section we shall discuss in more details two alternative representations of the acquaintance graph and give arguments for our choices.

The basic idea behind representing the CoT as a graph is motivated by the fact that this representation can efficiently serve as a basic data structure for the trust search algorithms we shall present in the course of this section. In this respect, we shall demonstrate that given a canonical ordering of the edges relating the nodes in G , our search algorithm can draw the graph incrementally in a greedy manner. In this context, we distinguish between two types of representations for the acquaintance graph:

Linear representation of the acquaintance graph

PGP Web of Trust [Zim94] can be viewed as a directed graph where the nodes are the keys, and the edges are the signatures. The path from P_1 to P_j can be found by tracing the set of the intermediate signed keys, such that for every keyring, it is very important that a node signs the key of other in order to be able to find such paths.

Similarly, in our linear representation of the graph, as illustrated in Figure 4.8.a, we first

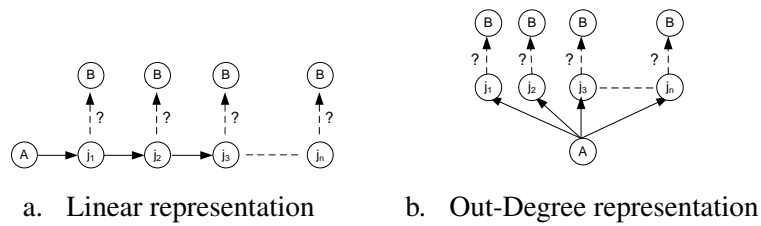


Figure 4.8: Representation of the trust acquaintance graph

seek to enumerate all nodes that are known to node P_1 , we follow a sequence of edges to *walk* through the graph, composed of pairs of nodes denoted (P_1, P_j) , until we reach a node which has a directed edge to node P_j . The path from node P_1 to node P_j is then a sequence of edges (P_1, P_2) , (P_2, P_3) , ... (P_{j-1}, P_j) , and its weight corresponds to the value of trust that has to be measured in function of the weight of each intermediate edge.

This graph, which is known as digraph [AB00], is quite simple as it has no loops and at most one edge between any pair of nodes, where we distinguish *out-degree* edges, the edges leaving a node, and *in-degree* edges, the edges entering a node. The presentation of this graph aims at keeping the search complexity of the algorithm linear, and the pointer does not necessarily have to go through all the nodes on the path; however, one can expect some intermediate nodes to contain links to the unknown node, and as soon as such a link has been found, the search path comes to an ending point.

From the performance perspective, this representation suffers from a known problem: The assumption for accepting a node P_j as trustworthy is estimated just for a given path and does not allow overall analysis for highly efficient distributed communities, because in most cases all participants need only check a small, local subset of the global trust graph.

Out-Degree representation of the graph

The limitations of the first representation of the graph motivate our investigations on the *Out-Degree* representation of the graph, which inherits from the theory of the Breadth-first graph. In this model, we illustrate our modified representation of the graph and show that significant improvements can be obtained using an exclusively oriented graph (a directed acyclic graph [Sen98]) by means of the edges leaving the node P_1 (out-degree) until reaching the node P_j .

As can be seen in Figure 4.8.b, the distance from node P_1 to node P_j is only based on the out-degree of the neighbors of node A that have to be visited as this graph representation is intended to serve as an overall assessment over all the existing nodes.

The path search works as follow: The pointer selects the first unvisited node, passes through its directed edge, and moves on in order to find the next unvisited node, always from the same starting node. These steps will be repeated until there will be no more unvisited nodes on the graph. Based on this feature, the trust search algorithms presented in the following section shall rely on this structure of the graph.

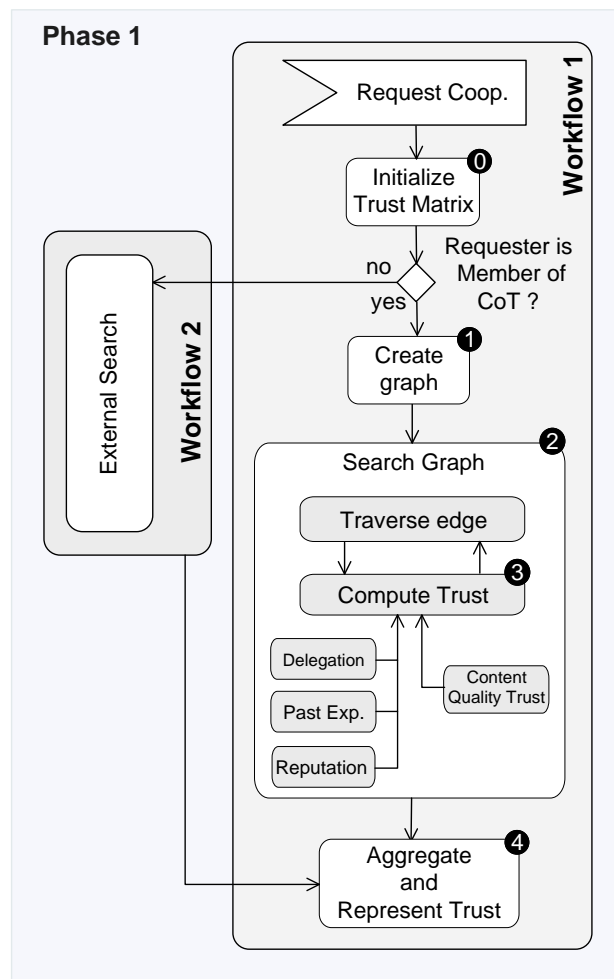


Figure 4.9: Workflow 1 - Trust assessment within the CoT

4.2.2 Trust Assessment

In this work, we proceed the establishment of trust as follow: Each of the participants in the CoT possesses an objective as well as an initial level of trust. This reflects, for example, the degree to which they will fulfill the assigned tasks or reciprocal obligations during collaborations.

In addition, the subject that communicates with a target participant (who might be outside the CoT) does not fully know this underlying trust value and must gather, during a process of the interaction, all available information from all possible sources in order to attribute by itself a level of trust and to decide about the collaboration party.

As we discussed in Chapter 2 through a wide number of scenarios and use cases, this type of collaboration requests implies a dynamic assessment of the prospective trust relationships. For this aim, we differentiate between two kinds of dynamic trust relationships:

- (i) Those among two principals that are both members in the CoT, but have not conducted business together beforehand.

- (ii) Those relationships crossing the borders of the CoT, i.e. those between a CoT member and an organization located outside the CoT.

In the following, we will present two workflows for dynamically setting up these two types of trust relationships. Key activities are illustrated by example.

As illustrated in Figure 4.9, our approach for dynamic trust assessment evolves through two main workflows: **Workflow 1** for managing dynamically the trust relationships among the participants inside the CoT, and **Workflow 2** for building and managing dynamically the trust relationships that cross the borders of the CoT.

The workflows are basically triggered by requests for cooperation directed at a CoT member who assumes the responder role. Workflow 1 is executed when the requester is known to the CoT; Workflow 2 is executed otherwise. Note that a considerable set of the activities of Workflow 2 build on the previous one.

4.2.2.1 Workflow 1: Trust assessment within the CoT

The target of workflow 1, which is based on five main activities (activities 0-4 in Figure 4.9), is to compute a trust value between two CoT members with regard to their end users. This computation is based on the trust information originating from trust dimensions and that is stored in the trust matrix.

This computation can be achieved in many different ways: Through inferences drawn from the outcomes of multiple direct interactions with these partners or through indirect information provided by others in the environment that have had similar experiences, have delegated rights or by reputation.

In the following, we shall detail the main steps constructing each activity:

Activity 0: Initialize the trust matrix

The workflows rely on the trust matrix introduced in Section 4.2.1.4. Therefore, before the trust relationships can be established from past interactions, the values $T_{l_{i,j}}(S)$ designating trust between CoT members need to be set to an initial value to reflect the agreements, for example, regarding identity sharing constraints and privacy policies. We use the tag *initial_trust* as an indicator of an initial weight of the trust relationships for the members when they join the CoT.

Note that the trust values need to be updated in the matrix after each new transaction taking place inside the CoT, thus enabling the members to raise or to lower trust values according to the update function we introduced in page 130. We shall address the issue of storing and updating the computed trust levels in phase 2 of our trust process model (in Subsection 4.3).

Beside initiating the trust relationships within a shared matrix in the CoT, the resources that, usually, are semantically described and stored in several nodes distributed in the CoT, have to be initiated and defined as well. In Phase 2 of our process model, we shall present the procedures we have undertaken for describing the resources in a unified manner in order to ensure a global accessibility from every node.

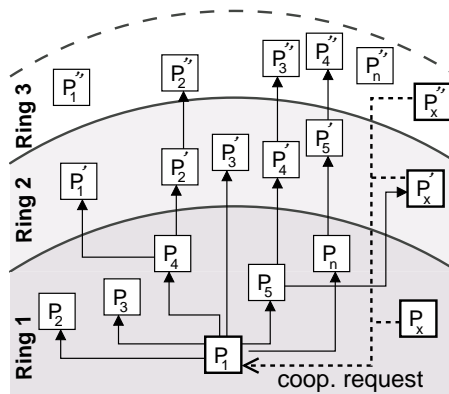


Figure 4.10: Trust graph

Activity 1: Create the acquaintance graph

Principals both inside or outside the CoT are represented as nodes in an acquaintance graph, with directed edges representing trust relationships relating them. The graph's structure relies on: (i) the $T_{l_{i,j}}(S)$ stored in the trust matrix for known nodes, or (ii) on the information provided by the neighboring nodes about the unknown node, with respect to a trust scenario S . In FIM environments this information can be usually collected by means of SAML-Assertions.

The resulting graph (Figure 4.10) may be divided into three *rings* reflecting members, direct neighbors and remainder (compare Subsection 4.2.1.5, Figure 4.7).

Activity 2: Breadth-first graph search

To find the trust relationship between requester and respondent, we progress breadth-first [THCS01] through the graph. The search, as illustrated in Algorithm 1 begins at the respondent P_1 (root node) in Ring 1 and searches outward the neighboring nodes whether one of them has a direct edge to node P_x . Function `getNeighbors` (line 7) returns 1 for those nodes that are connected to P_x , so that `getEdges` can read the weight of the corresponding edge ($T_{P_j P_x}$).

Next, the pointer traverses the graph recursively (the recursive call begins in line 18) and evaluates the trust values on the path between nodes P_1 and P_x by means of the function `computeTrust`, which will be explored in details in Activity 3. The search continues until the sought requester node P_x is found or until it fails to find an edge (when there are no neighboring nodes that are connected to P_x).

In the current case, i.e. when the requester is a CoT member, only the direct neighbors need to be assessed; they are identified by the function `getNeighbors`, as for this workflow, we need only to perform the search algorithm on the specified level where the CoT members are located (*Ring1* in Figure 4.10).

Space and time Complexity: Due to the fact that each node in this representation of the graph has either an edge to a parent or to a child node, it thus builds thus a tree, given n representing the total number of nodes in the tree. Following the Eulerian graph

Algorithm 1 Breadth-first search for requester P_x

Input parameters: Scenario S_i , set of trust relationships $(T_{P_1P_2}, \dots, T_{P_1P_n})$ to neighbors (P_2, \dots, P_n)

Output parameters: Trust relationship $T_{P_1P_x}$ from principal P_1 to principal P_x

```

1: begin
   /* Initialization */
2: for  $j = 2$  to  $n$  do
3:    $P_j := 0$ 
4:    $T_{P_jP_x} := 0$ 
5: end for
6: traverseGraph( $P_1, P_x, S_i$ )
   /* Check whether one of the neighbors has edge to  $P_x$  */
7:  $(P_2, \dots, P_n) := \text{getNeighbors}(S_i, P_1, P_x)$ 

   /* If so then read the weight of these edges */
8: for  $j = P_2$  to  $P_n$  do
9:   if  $j \neq 0$  then
10:     $T_{jP_x} = \text{getEdges}(j, P_x)$ 
11:   end if
12: end for

   /* If at least one edge exists, traverseGraph computes  $T_{P_1P_x}$  */
13: if  $((T_{P_2P_x}, \dots, T_{P_nP_x}))$  then
14:    $T_{P_1P_x} := \text{ComputeTrust}((T_{P_1P_2}, \dots, T_{P_1P_n}), (T_{P_2P_x}, \dots, T_{P_nP_x}))$ 
15:   return  $T_{P_1P_x}$ 
16: else
17:   for  $j = P_2$  to  $P_n$  do
18:     traverseGraph( $j, P_x, S_i$ )
19:   end for
20: end if

21: function  $\text{getNeighbors}(S_i, P_1, P_x)$ 
22: for  $j = P_2$  to  $P_n$  do
23:   if  $\exists(T_{jP_x})$  then
24:      $j = 1$ 
25:   else
26:      $j = 0$ 
27:   end if
28: end for
29: return  $(P_2, \dots, P_n)$ 
30: end

```

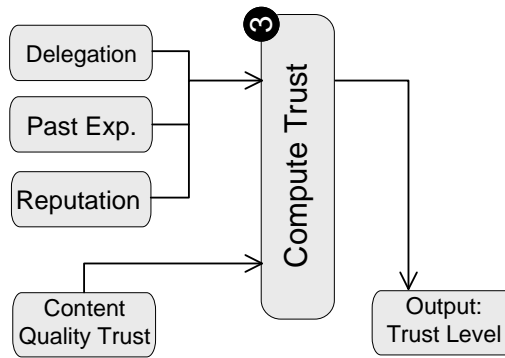


Figure 4.11: Function ComputeTrust

theorem [Kle97], the space as well as the time complexity of this tree is linear $O(n)$. This is because, for a given graph G , the relationship between the main parameters of the graph looks like the following: $v - e + f = c + 1$.

Where:

v is the total number of nodes

e is the total number of edges

f is the total number of faces

c is the total number of connected components

In the case of the recursive algorithm:

$v = n$

e is the total number of the edges, for which the complexity has to be evaluated

$f = 1$ as there is only one face

$c = 1$ there is always one connected component, because in the representation of the tree, any two vertices are located in the same connected component and there exists a path between them.

Therefore, $v - e + f = c + 1$ leads to $n - e + 1 = 2$. As a result $e = n - 1$, which implies that the complexity is $O(n)$. Keeping the space complexity of the search algorithm linear represents a considerable advantage for storing and retrieving the trust information from the trust matrix (the storage models of the trust information is discussed in Phase 2 in Subsection 4.3), especially in highly dynamic environments.

Activity 3: Computing trust values

The `ComputeTrust` function collates responses from multiple trust assessment invocations (delegation, past experience and reputation) into a single response. That is, this function operates on a set of different types of trust values and returns a single scalar value, identifying, thus, a trust level for a given identity in a given scenario.

As we will discuss in this subsection, this function could call a set of trust computation methods related to the well-known trust dimensions (Figure 4.11), each returning specific trust information, as this function may also be used in the select mode of meth-

ods and builds an aggregate structure containing all of this data following a specific representation.

In the following, we show conceptually how the `ComputeTrust` function is designed and evaluated, where the main principle is to pass several values as input to the function, and then perform aggregate procedures for the final representation of the resulting trust values.

4.2.2.2 Trust by delegation

As we discussed in the previous chapter, a large number of related works elaborate trust by delegation. One of the popular scenarios for building trust by delegation is that of Liberty Alliance Project in federated identity management. As we illustrated in the business example of the Liberty Alliance Project, in Subsection 3.1.2.1, a typical application that supports user single sign-on and delegation of rights enables the delegation of privileges within the CoT.

In these types of applications, services are usually combined in a business process, and act on behalf of a user. In this chapter trust by delegation in the context of delegation of rights for authentication purposes shall not be considered, due to the fact that this aspect has been widely investigated in other related works [KFJ01][BS04], more specifically in the Liberty Alliance Model (see Subsection 3.1.2).

In such systems, usually, the access manager within a given domain obtains and evaluates delegation and access permissions from other domains, in order to perform different operations such as read, write, save, and delete on the shared resources. Based on the results of the evaluation, the access manager allows or denies users the privilege of performing actions on the resources.

In this regard, trust by delegation can be established, when a principal from domain D_X needs to access any resource R provided by a resource owner P_R in another domain D_Y . The request from the principal P_x has to traverse the intermediate network before arriving at D_Y . On receiving the request, usually the certificate chain will be verified and if it is valid, P_x is allowed to access the resources. From this case, as shown in Figure 4.12, we notice that the trust from P_x to P_R is actually based on a delegation path including several intermediates (P_0, P_1, \dots, P_k):

$$P_x \leftarrow P_0 \leftarrow P_1 \leftarrow \dots \leftarrow P_k \leftarrow P_R$$

Apart from the delegation of rights for authentication, a user may additionally delegate some personal attributes to his proxy so that this proxy can use subordinate services according to the user's interests. The idea behind this type of delegation is to enable a set of attributes to be used as an authorization for using subordinate services. Therefore, disclosing attributes is thereby a delegation of access rights.

Due to the fact that the profile of the user arises at his proxy, the user cannot be sure whether his proxy follows its agreement according to the use of these attributes. Preventing misuse by looking at existing profiles means either to enable a user to control the use of disclosed attributes or to identify misuse and penalize the corresponding service provider afterwards. Both alternatives require the user's knowledge of the service's behavior, which cannot be assumed in advance nor generalized to other service providers.

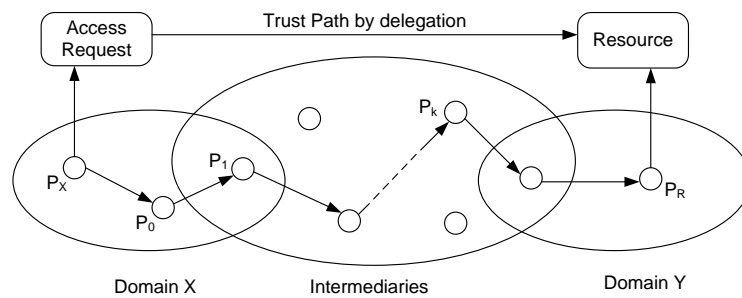


Figure 4.12: A process traversing domains and VO

In the next section we illustrate our solution for describing the behavior of interaction partners and build trust levels thereupon.

4.2.2.3 Trust from past experiences

Beside the alternative of building trust by delegation, in this subsection we present the second alternative, which aims at supporting partners in federated and collaborative environments to have a sort of accountability for assessing the behavior of each other.

As we discussed in the previous chapter, new paradigms in several related works have emerged for lightweight integration of enterprise resources. Among them we find Service Oriented Architecture, for example in the field of FIM and Grid Computing, as a unified way of describing, discovering and invoking resources in a heterogeneous and platform independent manner.

We have discussed that moving up in the paradigm from intra-enterprise to inter-enterprise integration of business resources results in the creation of virtual organizations (VOs). In VOs, obviously, a framework that properly controls and enforces the behavior of users (for example applications running on behalf of users) when using Grid resources, as well as the behavior of service providers is needed.

Quality of Service (QoS) parameters on the functional level as well as Key Performance Indicators (KPIs) on the financial and also technical levels, prove to be helpful for an organization to define and measure progress toward organizational goals in the CoT [Par07]. Precisely, ITIL Key Performance Indicators (KPIs) [Ste06] are used to assess if the ITIL processes of an IT organization are running according to expectations, so that resource owners and administrators are thus in a position to evaluate the quality of a collaboration, which in turn is the basis for the ongoing optimization and fine-tuning of the life cycle of the relationship.

In this regard, defining suitable QoS parameters and KPIs helps deciding what exactly is considered as *trustful action execution*. Additionally, the monitoring of these parameters not only affects the currently effective trust agreements, but also triggers a counter-measure automatically if some constraints are violated repeatedly. The selection of suitable QoS parameters and KPIs in the CoT, however, depends, among other things, on the possibilities to actually measure the indicators.

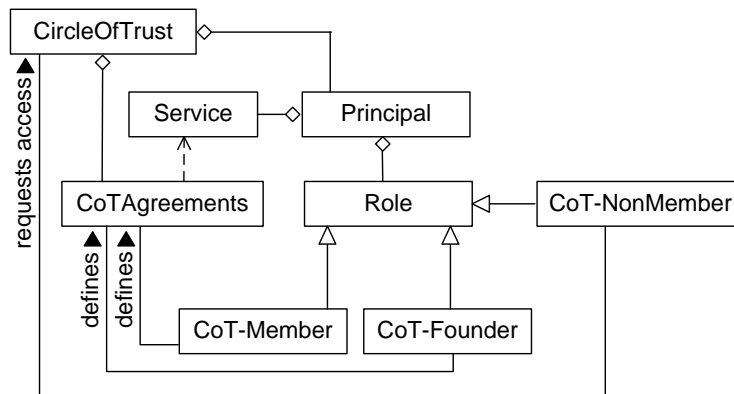


Figure 4.13: Agreements Rules in relationship with the CoT definitions

Identifying the performance indicators in the CoT

Such performance indicators and the corresponding measurement procedures, as introduced earlier, are important input for system requirements for the realization of the CoT. Unfortunately, these parameters are not always regarded with the same degree of importance among the collaborating organizations. An organization might consider the failure rate for a computing server as a Key Performance Indicator which might help the other partners selecting the computing services, whereas another organization might consider the percentage of income from return customers as a potential KPI.

Therefore, in the context of the CoT as a collaborative environment, it is necessary for every member (organization) to identify its performance indicators before engaging in the CoT. In the area of eContracting [GBW⁺98] [KGV00][Hof99] (as discussed in Chapter 3 in Section 3.5) and the area of Service Level Agreements (SLA) [LKDK02], we argue that the agreements among organizations can be extended with aspects of qualities of services (QoS) and performance.

We recall Figure 4.13 to illustrate the interdependencies with regard to the operational agreements and the CoT definitions. Accordingly, the solution of assessing trust from past experiences envisions that the agreements for identifying these parameters should mainly regard the following three categories of requirements:

- The CoT should possess a pre-defined collaboration process, eventually defined within the SLA in a form of an e-contract.
- Each participating organization or member should have clear goals and performance specifications and requirements for the collaboration processes. These specifications should include a unified description of their offered services and resources.
- The CoT should define unified quantitative and qualitative measurement methods for evaluating the results and for comparing them with the set of goals and commitments (set in the agreements).

Based on that, we propose a fine-grained description of the shared resources in the CoT, especially, on the level of the actions that might be performed on them. This description

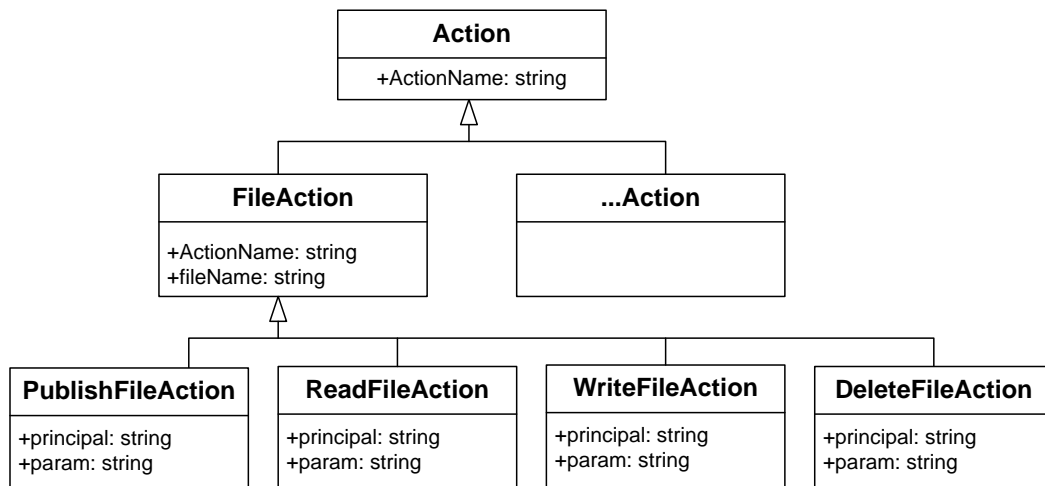


Figure 4.14: Fine-grained description of file storage actions

review suggests, in analogy with trust definition, that appropriate definitions of actions are highly context dependent. That is, by means of this resource description, the notion of history can be taken into account for reflecting the trustworthiness of the entity that performed the action. It shows, for example, how the execution of each action did change the state of the application.

Note that for this purpose, we focus on those performance parameters that relate to the behavior of the requester. Other performance parameters that relate to trust with regard to the quality of service and/or to the quality of resource content shall be addressed in detail in Subsection 4.2.3.

Example: File Storage (IntegraTUM Scenario)

As an example of resource description, we recall the IntegraTUM scenario (see Subsection 2.2.1), and analyze the possible actions for specific resources. For highlighting our solution on deriving trust from past experiences, we consider shared management of file storage among the content management repositories of the university partners.

Figure 4.14 illustrates our representation model, which parameterizes the actions for a given resource in types as well as subtypes. The subtypes of actions (e.g. publish, read, write, or delete) are in turn parameterized with additional parameters, which shall serve as key indicators for assessing the way the action has been executed on the resource and thus for quantifying trust from it.

For assessing trust about unknown principals, these parameters may represent any performance factors from which trust can be derived. In doing so, they facilitate the development and refinement of history hypotheses about how the principal shall be trusted in future interactions and whether the access can be granted or not. For example, it can then be ascertained whether the requester can be allowed to retrieve or delete the shared files.

As we can read from Table 4.1, the performance indicator parameters for *PublishFile*

Action	Performance parameter	Computation method	Possible outcomes
PublishFileAction	Availability	% Uptime (measured unavailability time)	(un)available
	Management Policy	% Number of incorrect grants and/or incorrect denials from total requests	Not conform to management policy

Table 4.1: Building trust from exemplary performance parameters for PublishFileAction

action for a guest lecturer, who tries to publish his file in the eLearning content management server, can be summarized in the following parameters:

- *Availability* represents the uptime of availability of the file in the server according to the collaboration policy. It could be measured as a percentage of the unavailability time from the total time as follows:

$$T_{client/server_\phi} = 100\% - \frac{\sum failure_\phi}{\sum verification_\phi}$$

Where the parameter ϕ in this case can be represented in the following way: $S : File : publishFile : Uptime$.

The given equation reflects the trust relationship between the publisher (the guest lecturer) and the storage server for a specific scenario S , which is represented in action $FileAction$ and its subtype $publishFile$. *Uptime* represents the performance indicator for reasoning about trust in this context.

Since we consider principally the failed requests from the total amount of interactions or verifications, we have to subtract the resulting percentage from 100% in order to get the desired trust values in the range of $[0, 1]$.

- *Management Policy* represents the statistics that show whether the management policies and responsibilities for publication and maintenance of files (for example period of publication, update deadlines, etc) have been respected as set forth. This parameter can be evaluated in the same manner as the *Availability* parameter:

$$T_{client/server_\phi} = 100\% - \frac{\sum PolicyViolate_\phi}{\sum interaction_\phi}$$

In the same manner the parameter ϕ represents: $S : File : PublishFile : Mgmt$.

A similar process for assessing trust by means of performance and behavior parameters is that for the action *deletefileAction*. When a server is asked to delete a file, in addition to checking the authorization and identity of the requester, it also examines their intentions to check how much they can be trusted when issuing commands of serious consequence, such as delete. The performance parameter *intention*, in this respect,

represents the percentage of the number of unauthorized delete requests from the total number of requests.

A more detailed exemplary representation of the possible actions, their outcomes, the performance parameters as well as the relevant computation methods are shown in Table 4.2. As we can deduce from this table, most of the performance indicators used for assessing trust of a client, who is interacting with a storage server, are self-explanatory, and have in common that they express a number of the failed or unauthorized actions as a fraction from the total amount of interactions and verifications. Based on that, we conclude that all these indicators can be computed as percentage of trust, at given point of time t_0 , according to the following equation:

$$T_{P_1 P_x \phi}(t_0) = 100\% - \frac{\sum failedRequest_{\phi}(t_0)}{\sum interaction/verification_{\phi}(t_0)} \quad (4.4)$$

ϕ represents in the general case $S : Res : Action : Param$.

In the same manner, this approach helps the consumer selecting services that better meet his needs based on the QoS attributes of the servers providing them. Table 4.3 shows some example of these.

Conceptually, basing the trust relationships and potential collaborations on the quality aspects is not an easy task because it faces the main challenge of open environments, where the quality of service (QoS) that a given service instance will deliver can not easily be predicted.

Although many works on QoS introduce metrics such as reliability, availability, and security, most of them, however, fail to give a full ontology of QoS in the context of trust in federated environments. Our approach is inspired by the work of Maximilien and al. ([MS04a][MS04b][Max05]), which specified a QoS ontology that enables to match services semantically and dynamically in Web Services. This semantic matching allows service agents, on the one hand, to match consumers to services using the provider's advertised QoS policy for the services, and on the other hand, the consumers' QoS preferences. The provider policy and consumer preferences are expressed using the concepts of a unified ontology.

On the basis of this approach, we adopt a similar representation of QoS parameters to dynamically capture data about service performance with respect to various QoS dimensions. These dimensions are obviously customizable and extendable according to the specification of the services and resources being shared in the environment.

Update of the trust values

While Equation 4.4 allows for different levels of initial trust to be established at a given point in time based on the assessment of past interactions, the different trust values mapped to scenarios might evolve in the course of future interactions in the form of $T_l(t) = T_l(t-1) \pm \Delta T_l$.

Therefore, an update function that takes care of the appraisal changes (increase of decrease defined in ΔT_l) of the trust values needs to be considered.

We recall the update function, presented in Equation 4.2 to formulate the update of both positive as well as negative interactions. The influence of this update on the trust level

Action	Performance parameter	Computation method	Possible outcomes
PublishFileAction	Availability	% Uptime (measured unavailability time)	(un)available
	Management Policy	% Number of incorrect or incompatible files from total published files	Not conform to management policy
ReadFileAction	Authorization	% Number of tries from total number of operations)	(un)authorized
	Honesty	% Number of unpaid bills from total transactions	payment not received
WriteFileAction (e.g. Update)	Authorization	% Number of unauthorized tries from the total number of operations	(un)authorized
	Carefulness	% Number of mistakes from the total number of operations	(un)intended
	Management Policy	% Number of incorrect grants or denials from total requests	Not conform to management policy
DeleteFileAction	Authorization	% Number of tries from total number of operations	(un)authorized
	Intention	% Number of wrong delete request operations from the total requests	(un)authorized

Table 4.2: Trust from past experience (Client)

Action	Performance parameter	Computation method	Possible outcomes
WriteFileAction (e.g. Update)	Availability	% Uptime (measured unavailability time)	(un)available
	Management Policy	% Number of incompatible files from total number of hosted files	Not conform to management policy

Table 4.3: Trust from past experience (Server)

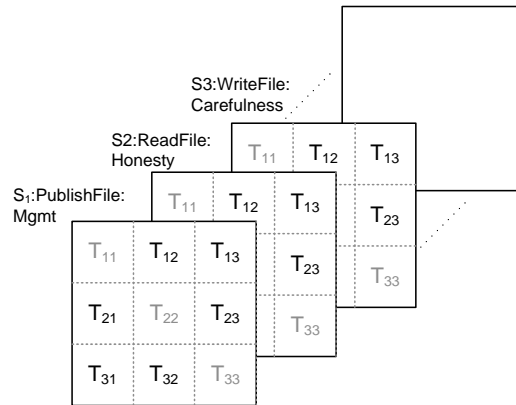


Figure 4.15: Representation of the trust values resulting from the audit system in the trust matrix

can be expressed as follow:

$$\Delta T_l = \begin{pmatrix} 1 - \frac{1}{2}e^{-\alpha(\sum interaction(\chi))} & \text{if } \chi = success_requests \\ 0 & \text{if } \chi = incomplete_requests \\ \frac{1}{2}e^{-\alpha(\sum interaction(\chi))} & \text{if } \chi = failed_requests \end{pmatrix} \quad (4.5)$$

From this equation, we distinguish between Failed Requests and Successful Requests, so that we can apply the functions defined in Formula 4.2 for both intervals by associating the test of $0 < interaction(\chi) < 0.5$ to Failed Requests, $0.5 < interaction(\chi) < 1$ to successful Requests and 0 for incomplete intractions, which might be considered as neutral.

Accordingly the trust level can be updated as follows:

$$T_{P_1 P_x \phi}(t) = \begin{pmatrix} T_{l_{P_1 P_x \phi}}(t-1) + \Delta T_l(\chi) & \text{if } \chi = success_requests \\ T_{P_1 P_x \phi}(t-1) & \text{if } \chi = incomplete_requests \\ T_{P_1 P_x \phi}(t-1) - \Delta T_l(\chi) & \text{if } \chi = failed_requests \end{pmatrix}$$

$$= \begin{pmatrix} T_{l_{P_1 P_x \phi}}(t-1) + \left(1 - \frac{1}{2}e^{-\alpha \sum_{j=t_0}^{j=t_1} succRequest}\right) & \text{if } \chi = success_requests \\ T_{P_1 P_x \phi}(t-1) & \text{if } \chi = incomplete_requests \\ T_{P_1 P_x \phi}(t-1) - \left(\frac{1}{2}e^{-\alpha \sum_{j=t_0}^{j=t_1} failedRequest}\right) & \text{if } \chi = failed_requests \end{pmatrix} \quad (4.6)$$

Representation in the Trust Matrix

In Figure 4.15, the resulting trust values from the audit system are illustrated in a similar manner as discussed in activity 0. In the resulting multi-dimensional matrix, we see that

Requirements	Fulfillment?
[Audit-Info]	✓ The audit information is described according to the key performance indicators and parameters.
[Audit-Metric]	✓ The estimated percentage represent the trust level in the interval of $[0, 1]$.
[Audit-Eval]	✓ The trust values are assessed according to Equation 4.4 and within workflow 1 shall be passed to the aggregate function. Note that the final evaluation, for access control, takes place in phase 3 (see Section 4.4).
[Audit-Storage]	? The fulfillment of this requirement shall be discussed in phase 2 in Subsection 4.3.

Table 4.4: Fulfillment of requirements of trust from past experiences

the different actions, subtypes of actions as well as the related performance parameters are mapped to scenarios, while the trust values among the principals represent two-dimensional squares of the matrix.

Fulfillment of the requirements for trust from past experiences

As stated earlier, the list of these parameters is obviously subject to extension and runtime update, as we shall discuss in Subsection 4.5. The performance parameters delineated above serve as an exemplary illustration of our approach for assessing trust from history-based monitoring. We conclude that this assessment provides a logical method that, by means of some key indicators, can describe to some extent the behavior of principals in the CoT and thus quantify the trust in them.

Moreover, dependencies among the actions that a principal is allowed to perform can be easily represented. In doing so, we can state, for example, that a given action is not allowed during the whole execution of the application. It is, however, allowed only when some other actions that depends on it have been already executed with an acceptable trust level (see Equation 4.4).

In this respect, we also conclude that this review highlights the need to test and extend pre-existing QoS parameters and performance indicators for shared services and resources in the CoT in order to ensure that they remain valid and relevant and retain their reliability in different settings.

Table 4.4 sums up the fulfillment of the requirements on trust from past experiences in light of this approach.

4.2.2.4 Trust by reputation

Conducting federated environments without the concept of CoT (i.e. with service consumers and service providers working individually) obviously limit the effectiveness of

the partners in judging the trustworthiness of each other. Consequently, it is helpful for them to share knowledge about behaviors as well as service quality statements. This knowledge description, however, should not be produced by the entity, whose trustworthiness is subject to verification, but instead should be provided by other entities that already interacted with it or had similar experiences.

From the previous discussions, we argue that the quantification of trust can be objective or subjective. While the objective quantification of trust is made automatically via audit mechanisms and tools, the subjective quantification of trust is usually achieved via some human agent. Both quantification methods have a validity period and need to be dynamically updated.

In the context of objective quantification, in the previous section, a logical history-based method for reasoning about trust from past experiences by means of performance and QoS parameters was presented. In this section, we focus on the subjective alternative for reasoning about trust by reputation.

Reputation management has come into wide use with the introduction of open computing and collaborative environments [GHP03], and in a form of exchanged ratings has proven to be very efficient for building trust. In this respect, exchanging ratings in the CoT enables the creation of a feedback loop by reporting on actions and opinions from other parties.

However, one important issue that one should not neglect is the fact that the ratings in the CoT can not be easily shared because they are based on the judgments of the various participants, which can be subjective. In this subsection we shall present our solution for assessing trust by reputation and reducing the effect of the potential arbitrariness of individual partners.

In previous works [BR07] [BD08] we have laid the foundation for a function for computing the trust level T , by integrating the aspects of reputation management, which involves the tracking of an entity's behavior within a federation and other entities rating that behavior. In this approach, the trust values are initiated according to activity 0 and updated with the rating values whenever a transaction between two partners happens and a feedback is given.

Algorithm 2 illustrates the function $\text{ComputeTrust} : P \times P \times S \mapsto T$ that computes the weight of the prospective trust relationship between two principals P_1, P_x that are connected by direct neighbors. Note that the edges of the relationships are based on the reputations values, which in turn can be assigned according to the same metric, we discussed in Subsection 4.2.1.2.

- In line 4, the algorithms checks if the principal P_1 with n neighbors already made a direct transaction with the new principal P_x in scenario S_i . If so, it ignores other weights and uses the weight of this relationship as its value of trust. This is due to the fact that if a line connects two nodes, they are considered to be *adjacent*, as the theoretical distance between two nodes in the graph is defined as the length of the shortest path between them [AB00].
- In the other case (line 6), i.e. when P_1 has not directly interacted with P_x (no direct edge is relating them), the value of trust is determined by a weighted average of the values for each of its neighbors with a direct edge to P_x . At the stage of line 12, the weight of the edges are then compared subsequently, in such a way

Algorithm 2 computeTrust: Compute $T_{l_{P_1 P_x}}$

Input parameters: Request of principal P_x , set of n principals (P_1, \dots, P_n) , set of trust relationships $(T_{l_{P_2}}, \dots, T_{l_{P_n}})$ to neighbors (P_2, \dots, P_n)

Output parameters: Trust relationship $T_{l_{P_1 P_x}}$ from principal P_1 to principal P_x

```

1: begin
2:  $S_i := \text{EvaluateRequest}(P_x)$ 
3:  $(\text{edge}, T_{l_{P_1 P_x}}[S_i]) := \text{DirectTrans}(P_1, P_x, S_i)$ 
4: if (edge) then
5:    $T_{l_{P_1 P_x}}[S_i] := T_{l_{P_1 P_x}}[S_i]$ 
6: else
7:    $(P_2, \dots, P_n) = \text{getNeighbors}(S_i, P_1, P_x)$ 
8:   for  $j = 2$  to  $n$  do
9:      $T_{P_1 P_j} = \text{getEdges}(P_1, P_j)$ 
10:     $T_{P_j P_x} = \text{getEdges}(P_j, P_x)$ 
11:   end for
12:    $T_{P_1 P_x}[S_i] := 0$ 
13:    $M := 0$ 
14:    $K := 0$ 
15:   for  $j = 1$  to  $n$  do
16:     if  $T_{l_{P_1 P_j}} \geq T_{l_{P_j P_x}}$  then
17:        $T_{l_{P_1 P_x}}[S_i] := T_{l_{P_1 P_j}}[S_i] \cdot T_{l_{P_j P_x}}[S_i]$ 
18:     else
19:        $T_{l_{P_1 P_x}}[S_i] := T_{l_{P_1 P_j}}[S_i]^2$ 
20:     end if
21:      $M := T_{l_{P_1 P_x}} + M$ 
22:      $K := T_{l_{P_1 P_j}} + K$ 
23:   end for
24:    $T_{l_{P_1 P_x}}[S_i] := \frac{M}{K}$ 
25: end if
26: function DirectTrans( $P_1, P_x, S_i$ )
27: if  $\exists T_{l_{P_1 P_x}}$  then
28:   directEdge := 1
29:   return (directEdge,  $T_{l_{P_1 P_x}}[S_i]$ )
30: else
31:   directEdge := 0
32:   return (directEdge)
33: end if
34: end

```

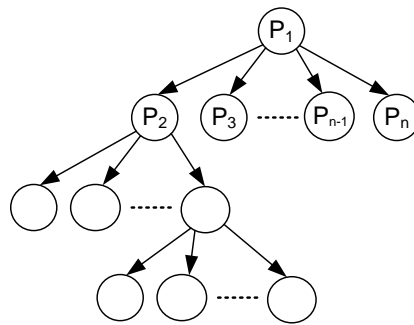


Figure 4.16: The breadth-first tree one gets when running the *ComputeTrust* on the given map by starting with P_1 to reach P_x

that it can be ensured that no principal can be trusted down the path more than any intermediary [GHP03].

- As demonstrated in Algorithm 1, this function shall be called recursively until there will be no node that might be connected to P_x down the line of each of the neighboring nodes (see Figure 4.16 which illustrates this search method).
- The successful search terminates when the result that an edge relating the requested node or one of its neighboring nodes to the requester node is found. Therefore, all successful searches terminate at an internal node. In contrast, the unsuccessful search terminates with an undefined status of the prospective trust level $T_{l_{P_1 P_x}}[S_i]$. The algorithm in this case simply assigns a value -1 to represent this status.
- The preceding analysis shows that in the case where all the neighboring edges have a highest weight $T_{l_{max}}$ (the highest level of trust in our trust metric is referred to be 1) to node P_x , without comparing the weight of the edges, $T_{l_{P_1 P_x}}$ will be:

$$T_{l_{P_1 P_x}} = \frac{\sum_{j=0}^n T_{l_{max}}^2}{\sum_{j=0}^n T_{l_{max}}} = \frac{n \cdot T_{l_{max}}^2}{n \cdot T_{l_{max}}} = T_{l_{max}}$$

In the same manner, in the opposite case, where all the neighboring nodes have the lowest weight $T_{l_{min}}$ to node P_x , following the comparison stated above, the value of $T_{l_{P_1 P_x}}$ is:

$$T_{l_{P_1 P_x}} = \frac{\sum_{j=0}^n T_{l_{P_1 P_j}}^2}{\sum_{j=0}^n T_{l_{P_1 P_j}}} = \frac{T_{l_{P_1 P_1}}^2 + T_{l_{P_1 P_2}}^2 + \dots + T_{l_{P_1 P_n}}^2}{T_{l_{P_1}} + T_{l_{P_2}} + \dots + T_{l_{P_1 P_n}}} = Cte$$

where $T_{l_{min}} < Cte < T_{l_{max}}$

Consequently, this formula states that for a finite subset of n nodes the sum of the trust values may be computed incrementally for estimating the trust level between two nodes only by using $O(n)$.

- Up to line 25 trust by reputation can be assessed in some quantitative real-values in relationship with a specified scenario. In some applications, a generalized trust level need to be associated to the principal. In this case, all the possible kinds of scenarios, usually typified in the communications and requests that might occur need to be taken into account. We explored how those trust-level oriented applications may be chained together, and we thought about capturing a simple average of the trust weight around the sum of those thinkable subject areas:

Let S_i be a positive integer and let S be a set of the scenarios in which a request for reasoning about trust might be sent. $T_{l_{P_1 P_x}}(S_i)$ represents the trust a principal P_1 has in P_x in a situation S_i ($S_i \in |S|$):

$$T_{l_{P_1 P_x}} = \lfloor \frac{\sum_{S_i=0}^S T_{l_{P_1 P_x}}(S_i)}{|S|} \rfloor$$

As a result, the algorithm estimates the level of trust by reputation of a principal, who is not directly connected with the requested principal through other intermediaries. If such intermediaries are found, then the algorithm returns the resulting value back specifying that there is a potential trust relationship for the requester to get access for the specified scenario. If no intermediary is found, as discussed earlier, the trust level shall be quoted as -1 indicating an unknown trust level.

Further, in most reputation management systems, the participants in a transaction are allowed to rate each other by submitting a comment and a rating about the quality of the transaction. The following subsection discusses this posterior phase of Algorithm 2, and shows how the rating of the transaction can be aggregated with the computed trust level.

The rating function (Update function)

Based on the given rating, obviously, the behavior of entities within and across the CoT changes and their reputations as well. It is therefore necessary that the approach of trust assessment by reputation takes these changes into account in order to adjust the trust level as follows: $T_i(t) = T_i(t-1) \pm \Delta T_i$.

In Chapter 2, we revised a number of reputation and trust metrics, which can be applied as a scale for the given ratings (objective or subjective). We brought forward the argument that a continuous metric (in our case $T_i \in [0, 1]$) best suits the requirements on trust in such fast changing and dynamic environments.

However, in contrast to the approach of assessing trust from past experience, where the trust values vary easily in the interval $[0, 1]$ in relationship with the quality of the interactions, for trust by reputation, we argue that such a continuous metric is not user-friendly. That is because the participant after a transaction often needs to express his opinion in a simple manner by giving a distinct rating level. See the example of the eBay rating portal in Figure 4.17, where the feedback left by members over a period of time vary between positive, neutral and negative.

Similarly, in this approach, we address the metric issue of the ratings by allowing the participant in a collaboration to rate the other correspondent according to these three levels:




Recent Feedback Ratings (last 12 months) ?			
	1 month	6 months	12 months
 Positive	84	341	739
 Neutral	0	1	2
 Negative	0	0	1

Figure 4.17: Feedback ratings in the eBay reputation system

- $\chi = 1$ for a positive rating
- $\chi = 0.5$ for a neutral rating
- $\chi = 0$ for a negative rating

Subsequently, we apply the update function (Formula 4.2) in order to scale these three levels of ratings in function with the total amount of interactions in the interval $[0, 1]$:

$$\Delta T_l = \begin{pmatrix} 1 - \frac{1}{2}e^{-\alpha(\sum interaction(\chi))} & \text{if } \chi = 1 \\ 0 & \text{if } \chi = 0.5 \\ \frac{1}{2}e^{-\alpha(\sum interaction(\chi))} & \text{if } \chi = 0 \end{pmatrix}$$

This final update on the trust level can be then performed according to Equation 4.8 as follows:

$$T_{l_{P_1 P_x}}(t) = \begin{pmatrix} T_{l_{P_1 P_x}}(t-1) + \Delta T_l(\chi) & \text{if } \chi = 1 \\ T_{l_{P_1 P_x}}(t-1) & \text{if } \chi = 0.5 \\ T_{l_{P_1 P_x}}(t-1) - \Delta T_l(\chi) & \text{if } \chi = 0 \end{pmatrix}$$

$$= \begin{pmatrix} T_{l_{P_1 P_x}}(t-1) + (1 - \frac{1}{2}e^{-\alpha \sum_{j=t-1}^{j=t} interaction(\chi)}) & \text{if } \chi = 1 \\ T_{l_{P_1 P_x}}(t-1) & \text{if } \chi = 0.5 \\ T_{l_{P_1 P_x}}(t-1) - \frac{1}{2}e^{-\alpha \sum_{j=t-1}^{j=t} interaction(\chi)} & \text{if } \chi = 0 \end{pmatrix}$$

Credibility and recentness of the ratings

Although reputation systems are emerging as one of the promising solutions for building trust, especially in the field of eCommerce among market participants, the influence of unfair ratings is a fundamental problem in reputation systems. Participants trying to get rid of their bad reputation history by adopting a new identity or two-party participants mutually improve each other reputations.

According to Zacharia and Maes [ZMM99][ZM00], a possible solution to these problems to safeguard the quality of the rating would be to make sure that:

- The ratings given by users with an established high reputation and long-termed membership in the system are weighted more than the ratings given by beginners or users with low reputations. This feature is enforced in our approach by performing the comparison between the weight of the edges connecting the principals directly or indirectly (see line 16 in Algorithm 2).
- Two users may rate each other only once. If two users happen to interact more than once, the system keeps the most recently submitted rating. That way artificially inflated reputations through a two-party collusion can be avoided.

Even if we allow each user to rate another only once, another way to falsely increase one's reputation would be to create fake identities and have each one of those rate the user's real identity with perfect scores. The damping function $\phi(R)$ from [ZMM99] would avoid both of these problems.

Finally, we have to consider the effect of the recentness of the ratings. We assume that recent data matters more in determining reputation and disregards very old ratings. In doing so, we ensure that the predicted reputation values are closer to the current behavior of the individuals rather than their overall performance, because the reputation values are associated with entities whose behavior might change over time.

For this aim, we just need to define a frequency of time after which the rating values may be overwritten periodically. Note that only ratings of the same context may overwrite other old ratings.

Example: Travel portal circle of trust

Recall the example given in Subsection 3.1.2.1. In this example, the CoT includes a travel portal in the role of identity provider *idp* and a group of hotels, airlines, and car rental agencies in the role of service providers *sp_i*. We start with a freshly initialized trust matrix, in which all members in the CoT get an initial trust value $T_{i,j} = initial_trust$. This assigned flag indicates that the requester is trusted initially by membership, which naturally might change in the course of the collaborations.

Obviously, the travel portal wants to offer the more appropriate financial options to its users by orienting them to reliable service providers. However, since it might be the case that it does not have a direct relationship with all hotel organizations; it needs a mechanism for selecting automatically the potential partner based on its reputation as well as previous experiences of other members in the CoT. The logical procedure, we presented in Subsection 4.2.2.3, is very helpful for evaluating the trustworthiness of the service provider for past experience, for example from performance indicators such as the reliability of the information posted on its web site, the response or booking time, etc. Note that all these parameters should be mapped to the so-called scenarios in the trust matrix.

In addition to that, Algorithm 2 enables the travel portal to assess trust from the reputation of the collaborating partner in the CoT (i.e. from self-given ratings with regard to more personal observations of other members). Taking care of the business policies and user preferences, it can thus select the most appropriate services based on their reputations.

The function `ComputeTrust` is used to compute the trust values and to determine the

best one for selecting one of the service provider for the scenario *Reliability*. The highest value the algorithm finds for this scenario is equal to 0.7. After 3 transactions with the selected sp_i , in case the travel portal finds his cooperation with sp_i satisfactory, it rates the sp_i with 1 for all the three transactions. Based on these ratings, the update function will be applied to scale this change and to raise the sp_i 's trust value $T_{l_{idp_1 sp_i}}$ from 0.7 as follows:

$$\begin{aligned} T_{l_{idp_1 sp_i} \phi} &= 0.7 + (1 - \frac{1}{2} e^{-0.1 * (\sum interaction_{\phi}(0.7)+3)}) \\ &= 0.7 + (1 - \frac{1}{2} e^{-0.1 * (44+3)}) = 0.877 \end{aligned}$$

Note that for this scenario the parameter ϕ represents $S : reliability$ and the parameter α is set to 0.1, allowing thus the change to increase or decrease within a small amount of interactions. $\sum interaction(0.7)$ is computed according to the inverse function of:

$$T_l = 1 - \frac{1}{2} e^{-\alpha(\sum interaction)} \text{ so that } \sum interaction = -\frac{2}{\alpha} \ln(1 - T_l).$$

Further, the resulting trust values from the reputation computation methods can be represented in the trust matrix in the same manner as for those values from past experiences.

As a consequence of this change, freshly computed trust values are stored in the matrix. Once one value has changed, all the other trust values will be recomputed accordingly. As illustrated in the example above, the trust value of the relationships between the selected service provider sp_i and other members in the CoT shall be updated automatically by `ComputeTrust`. Though we model the update and the extension of the matrix as an activity on its own (`Aggregate Trust`), the update of the trust values is in fact realized by the `ComputeTrust` function as well as the update function (see Algorithm 2).

Fulfillment of the requirements

In this subsection, we have demonstrated a collaborative reputation mechanism, which we designed to search, compute and establish reputation ratings among members in the CoT. Additionally, we presented an update function that can be used to scale the changes of the ratings. Table 4.5 discusses the fulfillment of the requirements on assessing trust by reputation.

However, we believe that incorporating reputation mechanisms in online communities may induce additional changes in the way users behave in the community. Further evaluation is of course required to measure the effects of the proposed mechanisms.

4.2.2.5 Workflow2: Trust assessment outside the CoT

Consider the case when a requester P_x outside the CoT, having already transacted a business relation with a CoT member, for instance with P_2 , wishes to cooperate with a CoT member, the respondent P_1 . This workflow investigates, whether it is possible to use the first relationship (P_x-P_2) to support the second one (P_x-P_1). As can be seen in Figure 4.18, Workflow 2 is initially based on Workflow 1. It begins with Activity 0 in

Requirements	Fulfillment?
[Rep-Value]	✓ Represented by the rating values χ (having three possible values 0, 0.5, and 1) scaled by means of the update function.
[Rep-Metric]	✓ End users use a discrete metric (0, 0.5, 1) while the final trust level by reputation is represented within the interval of $[0, 1]$
[Rep-Context]	✓ The ratings are assigned according to the specified scenario such as the QoS parameters.
[Rep-Cred]	✓ This requirement is fulfilled by applying the damping function from Zachario and al. [ZMM99].
[Rep-Recent]	✓ This requirement is fulfilled through the definition of an update period after which the ratings may be neglected and/or overwritten.

Table 4.5: Fulfillment of requirements of trust by reputation

which the request from P_x is verified. In such a request, the requester has to specify his credentials (e.g. public key) as well as the trust scenario (i.e. the service) in the focus of the request.

When performing the statement whether the "Requester is known in the CoT", i.e. if the requester is already recorded in the trust matrix and a direct edge between the principal P_x and one of the CoT members (e.g. in this case P_2) has been found, we continue with Activity 1 in Workflow 1.

To compute the desired trust relationships, activity 2 will then be applied recursively, however with the difference that this time the search goes this time beyond *Ring 1* (see Figure 4.7) and looks for potential edges in *Ring 2* as well (since P_x is outside the CoT).

In the opposite case no edge exists between a requester P_x and a CoT member. Hence, by definition, P_x is located in *Ring 3*. We handle this situation in Activity 5 by means of falling back to the authentication of the requester. Note that in this case no information about P_x 's past experience or reputation is provided.

Activity 5: Search by Certificates This function can be helpful to get more information about the requester when all the neighbors do not possess a direct reputation about him but have some indirect relationships via certificates. In this activity the following components are needed:

A value Set $C : \{C_1, C_2, \dots, C_n\}$ of possible certificates of principals issuing requests to access the CoT.

Cert_Search $P \times C \rightarrow T$ The request presented to the CoT that contains the requester public key, will be verified by *Cert_Search* to ensure this public key is signed by a third party *Certificate Authority*, who might be known to the CoT. This verification will be proceeded by means of the public key system used in the CoT,

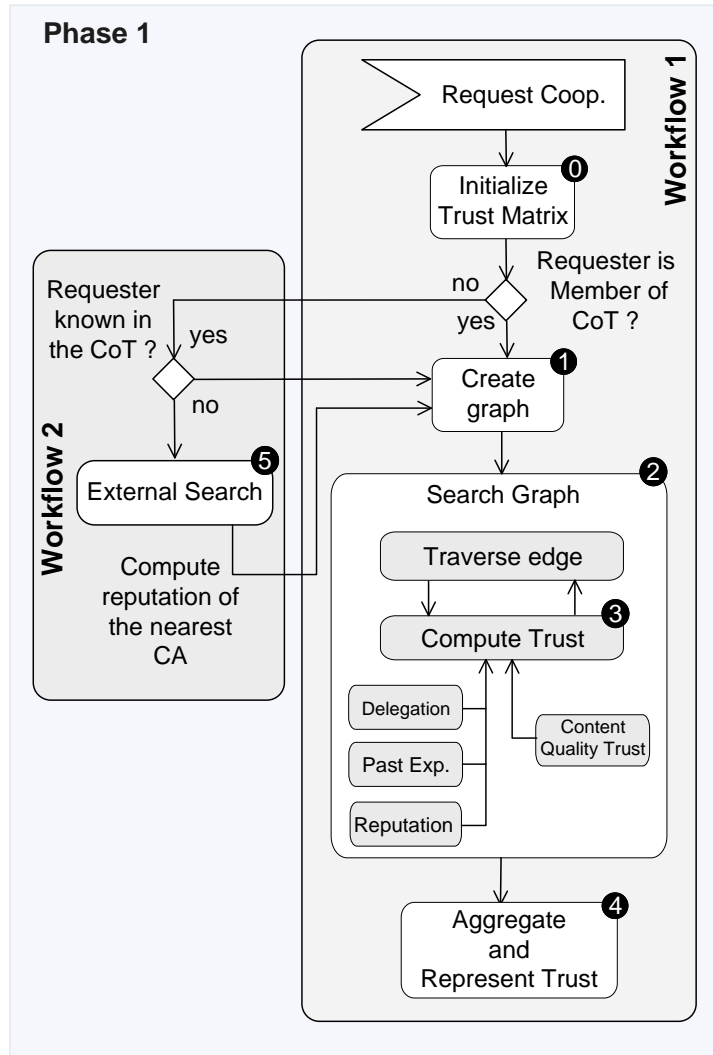


Figure 4.18: Workflow2: Requester is not a member in the CoT

which has a list of *trusted CAs* together with the corresponding public keys, so that the digital signature can be verified. Some CAs are so known that they are included by default in many public-key systems [X50].

The Algorithm 3 iterates until the algorithm identifies the nearest *CA* (in the certificate path) and computes its reputation. As a proper trust value pertaining to the principal P_x cannot be deduced, P_1 is provided with a confirmation of P_x 's identity, the identity of P_x 's nearest (in the key path) *CA*. The reputation of this *CA* may be computable by means of Workflow 1. Obviously, in this case the respondent has a much weaker basis for deciding about a cooperation with the requester.

Algorithm 3 ExternSearch: Estimate $T_{l_{P_1 P_x}}$

Input parameters: Set of trust relationships $(T_{P_1 P_2}, \dots, T_{P_1 P_n})$ to neighbors (P_2, \dots, P_n)

Output parameters: Trust relationship $T_{P_1 P_x}$ from principal P_1 to principal P_x

```

1: begin
2: for all  $P_{CA_i}$  such that  $P_{CA_i} \neq P_{CA_{root}}$  do
3:    $i := 0$  and  $j \in [1..n]$ 
4:   if  $\exists T_{l_{P_j P_x}}$  then
5:      $T_{l_{P_1 P_x}} := \text{ComputeTrust}(T_{l_{P_2}}, \dots, T_{l_{P_n}})$ 
6:     return  $(P_x, T_{l_{P_1 P_x}})$ 
7:   else
8:      $P_{CA_i} := \text{Cert\_Search}(P_x)$ 
9:     if  $\exists T_{l_{P_j P_{CA_i}(P_x)}}$  then
10:       $T_{l_{P_1 P_{CA_i}(P_x)}} := \text{ComputeTrust}(T_{l_{P_2}}, \dots, T_{l_{P_n}})$ 
11:      return  $(P_x, P_{CA_i}, T_{l_{P_1 P_{CA_i}(P_x)}})$ 
12:     else if  $P_{CA_i} \in (\text{Known\_CA\_List})$  then
13:        $\text{identified}(P_x)$ 
14:       return  $(P_x, P_{CA_i})$ 
15:     else
16:        $P_{CA_{i+1}} := \text{Cert\_Search}(P_{CA_i})$ 
17:     end if
18:   end if
19: end for
20: end

```

4.2.3 Content Quality Trust and QoS Trust

So far we have shown how our trust model reasons about trust from three different dimensions, namely, delegation, past experience, and reputation. The approach of computing trust in this model applies for requesters that are known in the CoT, e.g. by membership, as well as for external users who are not individually known in the CoT. These cases are enforced in two different workflows.

In this section, we study an additional way of reasoning about the trustworthiness of an entity, and address the challenges of dynamic partnership selection with regard to the quality aspects of the shared services and resources.

In Chapter 2, we highlighted the fact that the issue of building trust from the quality aspects presupposes a rich knowledge representations for services and qualities. Such representations prove to be helpful to capture the most important requirements for testing whether the involved parties in the cooperation provide resources and information as declared.

In the following, a method for representing the quality of services and resources in the CoT by means of measurable functional as well as non-functional aspects shall be presented.

4.2.3.1 Metric representation

While the metric we used for evaluating trustworthiness from past experiences expresses the frequency that the principal violates a particular constraint (expressed as a performance indicator and a QoS parameter with regard to entities behaviors), for content quality trust, however, other non security and behavior related parameters are also taken into consideration.

As we discussed in Section 3.6, a number of ontology representations and schemes for representing quality of services and quality of content resources exist. We argue that the types of ontology investigated in Vendatasubramanian et al. [VN97] and Maximilien et al. [MS04a] in the context of quality are considerably helpful to evaluate actions that are difficult to measure such as the benefits of a cooperation, engagement, satisfaction, etc. This consequently implies that each participant in the CoT specifies quality attributes for its services and resources, so that trust by quality depends on the capability of the participant of keeping those services at the requested quality levels.

An example of such a parameter for measuring the trustworthiness of a service is the reliability of the resource owner regarding the quality of the prospective shared resource.

According to Elvis Papalilo [Pap08], the measurement procedures for evaluating the QoS aspects can be divided into:

- **Quantitative procedures:** In these procedures the QoS parameters such as reliability, accessibility and availability, performance, responsiveness, etc. can be assessed by means of measurable values such as bandwidth, failure percentage and absolute times.
- **Qualitative procedures:** The second class of QoS parameters such as dependability, efficiency, flexibility, robustness, interoperability, security, etc are not expressed in absolute values but they are rather evaluated by means of some subjective evaluation aspects. ITIL KPI [Par07] also consider that financial aspects in a collaboration such as condition, credit, payment record, can contribute to the assessment of the trustworthiness of a subject by quality.

Actually both considered metrics should be described in the preliminary agreements between the participating organizations in the CoT (see Subsection 4.2.2.3). In doing so, the process for evaluating whether the party has not behaved according to the given metric specified in the agreement can be automated. Further, trust values based on the resulting behavior (positively or negatively evaluated) can in turn be assigned to the corresponding party.

4.2.3.2 Trust assessment procedure from quality aspects

The objective of this study for assessing trust from the quality aspects is not to investigate either quantitative or qualitative QoS ontology for FEs. However, the objective is to provide a methodology, which reasons about trust from existing QoS ontology schemes and requirements.

Therefore, this solution assumes that the overall assessment of the QoS is performed by the CoT participants themselves, since they are the ones able to evaluate the efficiency of the services offered and the grade of fulfillment of their requirements.

In this regard, our TBAC solution, through the monitoring of Quality parameters investigates more the process of deciding about issues such as what the participants are to be trusted for, what actions are they allowed to complete, etc.

Abstracting the common attributes for QoS parameters that express the quality aspects can be directly measured or broken up into measurable elements, in order to offer the possibility to create a history with data from past interactions among collaborating parties in the CoT.

A similar approach for assessing trust from past experience, as shown in Equation 4.4, can be applied here with the quality parameter related directly to the resources or service instead of actions and subtypes of actions. Equation 4.2.3.3 computes the trust level for a given point in time t_0 as a percentage of the failed requests when verifying the quality parameters from the total amount of resources that own the principal P_x .

$$T_{P_1 P_x \phi}(t_0) = 100\% - \frac{\sum failedRequest_{\phi}(t_0)}{\sum Resources(P_x)(t_0)} \quad (4.7)$$

Where ϕ represents $Res : QualityParam$.

In doing so, the establishment of a trust level in relationship with quality parameters helps to identify whether the collaboration parties were committed and the output of the shared services and resources was under the standards agreed upon, e.g. by evaluating the problems caused by errors, inaccuracies and imprecision in these shared data and resources.

The trust level can be updated in the same way as for *trust from past experiences* and *trust by reputation* according to the following equation:

$$T_{P_1 P_x \phi}(t) = \begin{pmatrix} T_{l_{P_1 P_x}}(t-1) + \Delta T_l(\chi) & \text{if } \chi = successRequest \\ T_{l_{P_1 P_x \phi}}(t-1) - \Delta T_l(\chi) & \text{if } \chi = failedRequest \end{pmatrix}$$

$$= \begin{pmatrix} T_{l_{P_1 P_x}}(t-1) + (1 - \frac{1}{2}e^{-\alpha \sum_{j=t-1}^{j=t} interaction(\chi)}) & \text{if } \chi = successRequest \\ T_{l_{P_1 P_x \phi}}(t-1) - \frac{1}{2}e^{-\alpha \sum_{j=t-1}^{j=t} interaction(\chi)} & \text{if } \chi = failedRequest \end{pmatrix} \quad (4.8)$$

4.2.3.3 Comprehensive example for Content Quality Trust

To demonstrate this approach, we recall the file storage scenario of the distributed eLearning platform that involves external lecturers and the content management server

of the eLearning platform. These types of shared platforms are characterized by dynamic user population, and by a very large amount of multimedia information, stored in a variety of formats and for different public categories.

In such systems, access policies are often specified based on resource descriptions, user qualifications as well as additional characteristics, rather than only on user identity verification. For example, aspects such as format, data versions as well as recentness of the revisions of the shared documents (which represent the resources in this case) in the eLearning platform can all be mapped to quality parameters.

The service agent finds services matching the desired quality, and then applies the consumer's policy on the available quality data to rank the service implementations.

Obviously such quality parameters are needed to clarify what constitutes trusting the owner of the shared resource and how the ranking of resource owners can be processed correspondingly. As it has been demonstrated in Equation 4.2.3.3, the ranking of the resources is computed from the quality-degree match, which is mainly based on what the provider advertises along with the interaction's results for the given quality, and the total amount of the resources it is providing in the CoT.

Therefore, the trust level of the principal P_x according to the quality parameter $VersionNr$ of the resource $PresentationFile$ in relationship with the total amount of the owned files in the eLearning portal can be computed as follows:

$$T_{P_1 P_x \phi}(t_0) = 100\% - \frac{\sum failedRequest(t_0)}{\sum Files(P_x)(t_0)}$$

ϕ represents: $PresentationFile : VersionNr$.

Fulfillment of the requirements

Table 4.6 sums up the fulfillment of the requirements that were investigated in Chapter 2 within the approach of trust assessment with regard to quality aspects of the content of the shared resources in the CoT.

4.2.4 Aggregation between the three dimensions of collaboration trust

In the previous sections, different logical computation methods have been presented for the computation of the trust level for a potential principal. These computations are based on the well-known trust dimensions (trust by delegation, trust from past experiences, and trust by reputation).

In this section an aggregation algorithm that analyzes and aggregates the different trust levels, which might result from the different computation methods, shall be presented. This algorithm basically considers situations, when, for example, information on previous behavior, content quality or reputation feedback about the requester all exist. In this case, the requester principal might obtain trust levels for the same scenario. These trust levels might be unequal and even contradictory. Therefore, an aggregation mechanism, which assists the requested principal for reasoning about this information in order to decide how much trust should be finally put on the requester, is obviously needed.

The intuition behind the Aggregation Algorithm is the following:

Algorithm 4 Aggregation Algorithm: Estimate $T_{l_{P_x}}^{final}$

Input parameters: Request of principal P_x , Scenario S_i , set of trust levels $(T_{l_{P_x}}^{past}(S_i), T_{l_{P_x}}^{reputation}(S_i), T_{l_{P_x}}^{content}(S_i))$

Output parameters: Trust level $T_{l_{P_x}}^{final}$

1: **begin**

2: $S_i := \text{evaluateRequest}(P_x)$

3: $(T_{l_{P_x}}^{past}(S_i), T_{l_{P_x}}^{reputation}(S_i), T_{l_{P_x}}^{content}(S_i)) := \text{ReadMatrix}(S_i)$

/* The case where the trust level is available from a single dimension */

4: **if** $((\exists T_{l_{P_x}}^{past}(S_i)) \text{ and } (\nexists T_{l_{P_x}}^{reputation}(S_i)) \text{ and } (\nexists T_{l_{P_x}}^{content}(S_i)))$ **then**

5: $T_{l_{P_x}}^{final}(S_i) = T_{l_{P_x}}^{past}(S_i)$

6: **end if**

7: **if** $((\nexists T_{l_{P_x}}^{past}(S_i)) \text{ and } (\exists T_{l_{P_x}}^{reputation}(S_i)) \text{ and } (\nexists T_{l_{P_x}}^{content}(S_i)))$ **then**

8: $T_{l_{P_x}}^{final}(S_i) = T_{l_{P_x}}^{reputation}(S_i)$

9: **end if**

10: **if** $((\nexists T_{l_{P_x}}^{past}(S_i)) \text{ and } (\nexists T_{l_{P_x}}^{reputation}(S_i)) \text{ and } (\exists T_{l_{P_x}}^{content}(S_i)))$ **then**

11: $T_{l_{P_x}}^{final}(S_i) = T_{l_{P_x}}^{content}(S_i)$

12: **end if**

/* The case where the trust level is available from more than one dimension */

13: **if** $((\exists T_{l_{P_x}}^{past}(S_i)) \text{ and } (\exists T_{l_{P_x}}^{reputation}(S_i)) \text{ and } (\nexists T_{l_{P_x}}^{content}(S_i)))$ **then**

14: $T_{l_{P_x}}^{final}(S_i) = \text{aggregatePastRep}(T_{l_{P_x}}^{past}(S_i), T_{l_{P_x}}^{reputation}(S_i))$

15: **end if**

16: **if** $((\nexists T_{l_{P_x}}^{past}(S_i)) \text{ and } (\exists T_{l_{P_x}}^{reputation}(S_i)) \text{ and } (\exists T_{l_{P_x}}^{content}(S_i)))$ **then**

17: $T_{l_{P_x}}^{final}(S_i) = \text{aggregateRepContent}(T_{l_{P_x}}^{reputation}(S_i), T_{l_{P_x}}^{content}(S_i))$

18: **end if**

19: **if** $((\exists T_{l_{P_x}}^{past}(S_i)) \text{ and } (\exists T_{l_{P_x}}^{reputation}(S_i)) \text{ and } (\exists T_{l_{P_x}}^{content}(S_i)))$ **then**

20: $T_{l_{P_x}}^{final}(S_i) = \text{aggregatePastRep}(T_{l_{P_x}}^{reputation}(S_i), T_{l_{P_x}}^{content}(S_i))$

21: **end if**

22: **function** $\text{aggregatePastRep}(T_{l_{P_x}}^{past}(S_i), T_{l_{P_x}}^{reputation}(S_i))$

23: $T_{l_{P_x}}^{final}(S_i) = T_{l_{P_x}}^{past}(S_i) + \text{Update}(T_{l_{P_x}}^{reputation}(S_i))$

24: **return** $T_{l_{P_x}}^{final}(S_i)$

25: **function** $\text{aggregateRepContent}(T_{l_{P_x}}^{reputation}(S_i), T_{l_{P_x}}^{content}(S_i))$

26: $T_{l_{P_x}}^{final}(S_i) = T_{l_{P_x}}^{content}(S_i) + \text{Update}(T_{l_{P_x}}^{reputation}(S_i))$

27: **return** $T_{l_{P_x}}^{final}(S_i)$

28: **end**

Requirements	Fulfillment?
[Content-Quality]	✓ In the current subsection a procedure for mapping quality parameters to the given shared resources in the CoT has been demonstrated, in such a way that a trust level can be assigned to the owner of the resource in relationship with content quality.
[Content-Rep]	✓ The same procedure for computing trust by reputation of a principal applies when exchanging rating feedback about the content of the resources. Therefore, the fulfillment of this requirement is identically handled by the approach of trust by reputation, except that the quality parameters are bound directly to the resource instead of the actions and subtype of actions.
[Store-Complex] [Store-Monitor] [Store-Conflict]	? The fulfillment of this requirement shall be discussed in phase 2 in Section 4.3.

Table 4.6: Fulfillment of requirements of Content Quality Trust

- When an access decision is requested, the algorithm evaluates this request, and by means of the specified scenario loads from the matrix all available computed trust levels, which has been computed from the available computation methods. This is handled by the function `ReadMatrix`.
- If just one single trust value is available for the given scenario, as stated in lines 5, 8, and 11, this value shall be considered as the final trust level, in such a way that no aggregation is needed.
- In the opposite case, where the trust values exists from more than one trust dimension, an aggregation rule is needed. The algorithm in this case attempts to match the given values in a single final value.

While the computation of the trust level from past experience is performed automatically (after each interaction) by means of the monitoring tools, the trust level by reputation can be assessed only if at least one interaction partner leaves a rating level at the end of the interaction.

Due to this fact, the aggregation function `aggregatePastRep` sets up $T_{l_{P_x}}^{past}(S_i)$ as the starting value and increments it or decrements with $T_{l_{P_x}}^{reputation}(S_i)$ according to the update function presented in subsection 4.2.1.2. The same reasoning applies for the aggregation between $T_{l_{P_x}}^{content}(S_i)$ and $T_{l_{P_x}}^{reputation}(S_i)$ within the function `aggregateRepContent` in line 25.

- Next, potential aggregation between $T_{l_{P_x}}^{past}(S_i)$ $T_{l_{P_x}}^{content}(S_i)$ is intentionally not considered for the following reason: Previously we have demonstrated that the trust level from past experience can be represented in the form of $T_{l_{P_1 P_x}}^{past}(Res :$

Action : Param), while the trust level with regard to the quality aspects of the resources in the form of $T_{l_{P_1 P_x}}^{content}(Res : QualityParam)$.

From these representations, it can be deduced that trust by quality is just an instantiation of trust from past experience by defining an action `own` indicating thus the ownerships of the resource whose quality is subject of trust. Accordingly, the trust level can be represented in the form of $T_{l_{P_1 P_x}}^{content}(Res : Action : QualityParam)$, with *Action* always equals to `own`.

Based on that, we argue that if $T_{l_{P_x}}^{past}(S_i)$ and $T_{l_{P_1 P_x}}^{content}$ match the same scenario *Res : Action : Param* they can be considered as identical, so that no aggregation is needed.

Fulfillment of the requirements

In this section, trust level aggregation process in which information about trust levels is gathered and expressed in a summary form, has been presented. The common aggregation purpose in this respect is to get more trust information about a particular principal based on specific dimensions such as past experience, reputation, and content quality. The final trust level about the prospective principal can then be used for choosing reliably trustful interaction partners.

In Table 4.7 the fulfillment of the requirements that fall in this category shall be discussed.

4.3 Phase 2: Storage and management

While phase 1 of the trust process model focused on the dynamic assessment of the trust levels, as illustrated in Figure 4.19, phase 2 shall consider issues that are related to the distribution, the storage, and the management of the resulted trust information among the involved partners, which usually are located in different administration domains in the CoT.

In this regard, it is important to observe that in such an open environment a storage system for trust management has first and foremost to take into account the properties of data basis, the data structure, and in particular the access and privacy policies of the underlying system. For example, the trust layer has to take into account, that not all data is equally accessible when assessing trust based on statistical evidence derived from transaction data.

4.3.1 Organizational models

As discussed earlier, most effective and affordable strategies for developing a system of digital archives for trust management is to decide between using a completely distributed, or a centralized structure, or a combination of aspects from both structures. This choice is extremely relevant for issues of collecting digital trust information, protecting their integrity over the long-term, and retaining them for future use.

While centralized structures have proven to be very cumbersome and unsuitable in open and distributed environments, a critical issue in the case of distributed digital archiving

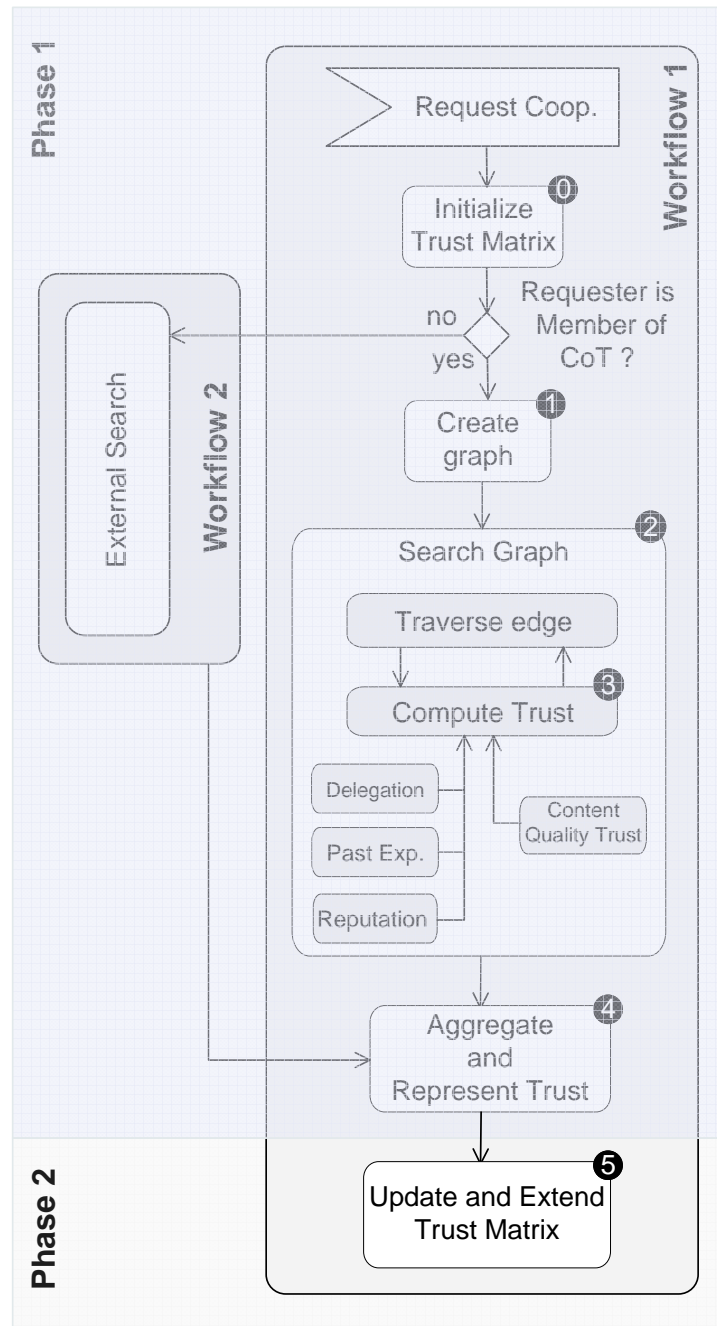


Figure 4.19: Phase 2: Represent, update and store the trust information

Requirements	Fulfillment?
[Aggre-Collect]	✓ The data structure of the trust matrix helps the computed trust levels to be collected and provided to the aggregation algorithm.
[Aggre-Scheme]	✓ The aggregated trust levels are represented in the same manner following the same scheme, such as they express the scenarios of the trust relationships and are ranged in the same interval of $[0, 1]$.
[Trust-Interm]	✓ They may be assessed from the judgments of TTP as well as from the experience of the requested principal itself.
[Trust-Level]	✓ The final trust level is estimated by the aggregation algorithm.
[Trust-Metric]	✓ Since all the computed trust levels are expressed by the same metric ($T_i \in [0, 1]$), the final trust level respectively follows this metric.
[Trust-Context]	✓ In analogy with the trust metric, collected in the so-called [Trust-Policy] are enforced within the aggregation algorithm.

Table 4.7: Fulfillment of requirements of aggregation and final representation of the trust level

infrastructure is to assure the existence of a sufficient number of trusted organizations capable of storing, migrating, and providing access to digital trust information.

In Section 3.1.2.3, three classes of the Liberty Alliance organizational models have been presented, where an organization or a set of organizations play the role of the founder entity that is responsible for managing and archiving the trust information within the CoT. In these models the trust information is mainly represented in the Trust Anchor Lists (TAL) as well as the Business Anchor List (BAL) that enable indirect trust relationships to be built among distinct principals.

In this work, the idea behind the founder of the CoT represents the starting point of our organizational model – as demonstrated in previous work in [Bou07] – for managing the storage of the trust information. That is, in the creation phase of the CoT, this model requires that a founder principal is identified as a trusted source for assuring the longevity of information. Although the founder as a central entity is defined in the role of an *administrator*, this model does not mirror a typical centralized architecture as it includes important features of a decentralized management structure. This is due to the fact that all the participants in the CoT have the right to update their own trust data and can access data from the neighboring participants.

4.3.2 Data structures

The investigation on the organization and logical concepts of the data structure for the storage of the computed trust levels is a very important matter to assure that it can be

used efficiently. That is, because a carefully chosen data structure will allow the search as well as the computation algorithms with critical operations to be performed, using as few resources, such as execution time and memory space, as possible.

Usually, the choice of the data structure begins with the choice of the data type on the one hand, and the access control policies and privacy matters on the other hand.

In this work, for the purpose of storing the trust levels in a multi-dimensional matrix, a linear data structure such as a matrix representation or combined lists are particularly well-suited for design and implementation considerations. The trust matrix shall then be represented as a set of tables that contain entries, which may be numbers as $T \in [0, 1]$ on which numerical operations such addition and multiplication may be performed.

Once the data structure of the trust matrix is chosen, the algorithms to be used on it become relatively obvious. As we shall demonstrate in Chapter 5, this representation enables to keep track of the entities behaviors and to record data that depend on the parameters that describe the scenario of the interaction.

As it is shown in Figure 4.19, the storage of the trust information in Activity 5 regards either updates on existing trust information (for example updates performed by the update function) or extensions of this information, such as extension with the storage of new trust relationships of entities that recently entered the CoT, as well as extension with regard to new cooperation scenarios.

- **Update of the trust information:** For this objective, the storage system (e.g. the storage engine) by means of the monitoring tools receives instantaneous updates or the freshly assigned rating feedbacks after the processing of an interaction. These updates shall subsequently be reported in the trust matrix, reflecting thus the CoT members' experiences with other partners.

Obviously, before overwriting existing values in the trust matrix, these new updates first need to be logged into buffers, by maintaining a counter or other update indicators that represent each storage subpart. Subsequently, for each update request, the storage system determines the type of the update that may be performed as well as the related policies to assure whether target subpart's update activity does not violate prescribed rights.

In doing so, especially, by applying the aggregation algorithm (presented in Subsection 4.2.4) before the storage activity, it can be assured that the storage conflicts discussed in the requirements analysis may be avoided.

- **Extension of the trust information:** Extensions on the trust information represent either (i) addition of new trust relationships when a new member enters the CoT or an external principal interacts with one of the members, or (ii) enhance functionality of the main information features, by creating a new table in the matrix, which represent a new service or simply a new cooperation scenario. Figure 4.20 illustrates the two update types.

Extensions are one of the main advantages of this phase, as they give CoT administrators and CoT members the ability to adapt the collaboration process in the CoT to their new services and requirements.

The storage model should be monitored at each organization side, in such a way that for each collaboration type, a number of useful trust monitoring and evaluation parameters

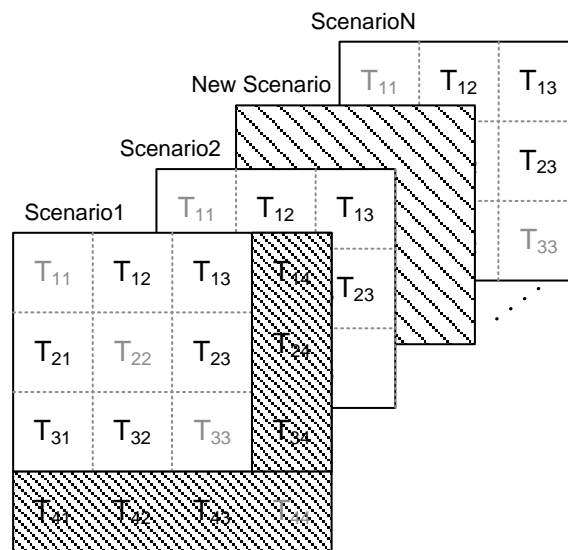


Figure 4.20: Possible updates and extensions on the trust matrix

can be defined.

Table 4.8 recalls the requirements on the storage issues and summarizes their fulfillment through the presented approach.

Beside investigations on data structures and data types, the privacy constraints for managing the entries in the trust matrix pose a crucial issue in federated environments. However, since we start from the principle that the collaboration effort should complement rather than replace existing local security and privacy policies and constraints, in this work, we address merely the privacy measures that concern the computed and distributed trust levels.

We shall detail the enforcement of the privacy matters on the storage of trust information in Section 5.4.1.

4.3.3 Risk managements aspects

As discussed previously, a principal's trust level changes over time, typically based on feedback and recommendation mechanisms known from audit systems or reputation management; however, service providers as resource owners in the CoT must always consider the risk of granting resource access to previously unknown users and cannot afford to rely solely on vague trust recommendations, especially because several reputation management approaches that were used in eCommerce environments turned out to be bogus or susceptible to fraud.

That is, since trust is calibrated to the potential risk if the other party does not act as expected, mechanisms that outweigh the risk of incidents caused by malicious or uncaredful users are required.

Conceptually, due to the great variety of trust and risk metrics available on both, the algorithmic and the management level, we first define the terms and data structures we use throughout our work. We then discuss the workflows we use for the quantification

Requirements	Fulfillment?
[Audit-Storage]	✓ As for the trust information by reputation, we have shown that the trust levels extracted from the audit data can be stored according to the same data structure and type of the trust matrix.
[Store-Complex] [Store-Monitor] [Store-Conflict]	✓ By means of the aggregation algorithm, the monitoring tools as well as the scenario-separated-alike storage model, conflicts which may occur when storing information about the trustworthiness of identities against that of resources' content can be managed and handled.

Table 4.8: Fulfillment of requirements on the storage of the trust information

of trust levels and risks.

Risk quantification

Resource owners must specify the risk levels of their resources. Given a risk calculation algorithm *risk*, the resource's risk level γ depends on the action to be performed at a certain point in time on the resource, but is independent of the user: $\gamma_{(r,a,t)} = risk(r, a, t)$

In real-world scenarios, each organization must define its own risk level assignment rules. Generally, they are based on legal and regulatory compliance responsibilities, the SLA impact if the resource federation rules are not met, and the threats resulting from unauthorized access.

This phase highlights the incorporation of trust as well as risk levels in the access control policy as a viable solution to the problem of access control in open collaborative environments, where decisions of an autonomous nature need to be made based on information and evidence.

The next phase (validation phase) seeks to investigate how the concept of trust and risk levels can be used in the access control policy in conjunction with the identity attributes of users.

4.4 Phase 3: Validation

The validation phase, which follows directly Phase 2 of the trust process model as illustrated in Figure 4.21, provides background to the progressive role that trust plays in access control decisions. Based on the scenarios discussed throughout this thesis, Phase 3 illustrates how access control decisions can be made autonomously by including a trust level of the requestors in an access control policy.

Including the notion of trust level in access control across organizations facilitates to

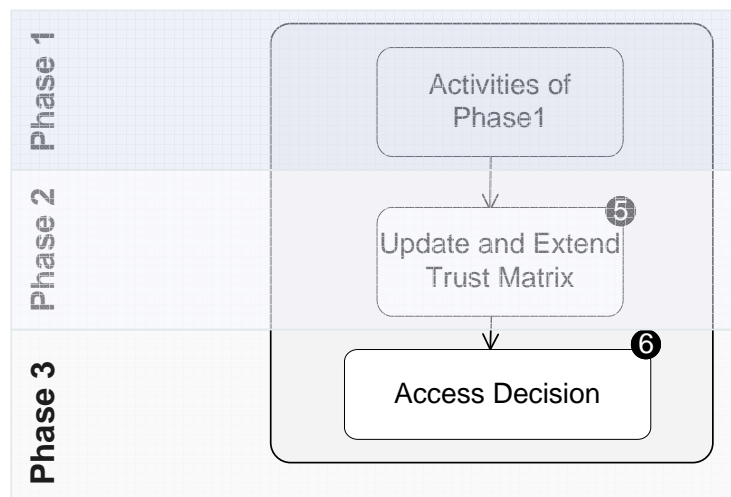


Figure 4.21: Phase 3: Validation

determine, for example, for each action that the requester tries to perform, whether this action is included in the current set of allowed actions that may be performed automatically when, beside other conditions, a certain level of trust is reached.

In this regard, the access policy can be expressed in a way such as a high degree of trust in a participant would mean that it is likely to be chosen as an interaction partner. Conversely, a low degree of trust would suggest that the participant is no more selectable, especially in the case when other, more trusted interaction partners are available.

Example: An application example would be that presented in Subsection 4.2.2.3 for file storage with different possible actions that may be performed in a distributed manner. For the action *readFile*, imagine a storage service, where retrieving files does require payment. The requester (external student) submits a request to read a certain file. The server weighs up the trust level given to this principal and its confidence in the authenticity against the importance placed in the confidentiality of the file.

This verification is usually defined within a policy on the storage server's side, so that for example, for a public presentation, only a low trust level (it must be however greater than the neutral value) is needed for granting the access, while for a presentation intended only for project partners, the server may decide the file is sufficiently important to expand resources to those principals that only possess a high trust level in addition to further authenticity and confidence verifications.

4.4.1 Establishment of Trust Agreements

The mission of the interorganizational trust agreements is basically to develop agreements, including data, resources, share and duties agreements that will permit the participating CoT members and their related departments and entities the possibility to conduct interoperable information exchanges.

Fine-tuning the privacy and security components of the agreements is usually the pri-

mary focus. Once the trust agreements are drafted and established, they shall be deployed (that is, translated into executable security and privacy rules) to enforce the interorganizational security and privacy policies. Moreover, it is important that none of the trust agreements compromises or conflicts with an existing individual organization's policies and constraints. The collaboration effort should complement rather than replace existing local security policies and constraints.

4.4.1.1 Challenges

Unfortunately, establishing standards and collaboration models for interorganizational agreements is a complex process that poses several challenges and potential obstacles, including for example:

- Variety of purposes of the collaborations in the CoT as well as variations in internal privacy and security policies between organizations.
- The consent policies, which represent the level of the consumer's participation, role, influence and control over the personal information exchanges and over the trust information disclosed to other entities across the CoT borders.
- The organizational model of the CoT, such as to differentiate between relationships of one to one vis-a-vis one to many vis-a-vis many to many relationships between the organizations.
- Level of granularity of the security protocols that the CoT should enforce, with respect to privacy policies and security standards (secure connection, secure transmission, encryption, identification, authentication, authorization, audit controls, non-repudiation, integrity, etc).
- Appraisal of the actions that need to be taken by parties in case of inappropriate use and disclosure of information, or in case of other breaches of agreement.

4.4.1.2 Alternative solution

It is important to note that the aim of this thesis with regard to the challenges, presented above, is not to develop and to investigate foundational reference guidelines that describe or compare the requirements mandated by every organization in this concern, since these types of guidelines vary in function of the collaboration and circle of trust scenarios, the aim, however, is to establish a model for identifying, representing and resolving the policies set by each member in the CoT with regard to information disclosure consent and requirements.

In general, the higher the degree of assurance required, the more inflexible is the system enforcing it. However, at the other end of the scale, in some collaborative environments even more flexibility is demanded, while in others, privacy of the data represents a major concern.

For instance, in military scenarios, where secrecy needs to be guaranteed at all costs, the CoT infrastructure has to use a rigid access control system to enforce data-usage policies. Whereas, for medical applications and public health scenarios [LLT00] more flexible systems are needed which guarantee privacy of patients without interfering

with the availability as a high-priority treatment of data, by allowing users to override confidentiality permissions.

Accordingly, by clarifying and documenting consent requirements and agreements, the TBAC framework can thus enable increased and automated interorganizational electronic information exchange to some extent. In the following preliminary measures that are to be taken into consideration by the CoT during planning as well as validation phase are listed:

- Beside the global policies imposed by the concept of the CoT, each participating member has the possibility to express and enforce additional local policies. That is, each member collects and submits to the CoT founder its privacy constraints and policies regarding information sharing and disclosure.
- Standards, which provide guidance on data protection requirements to facilitate the transfer of personal data across organizational borders are obviously needed. As a solution for the previously discussed issues on sharing the trust information (trust levels including its different dimensions) in the CoT, we propose to extend the existing guidelines with policies that specify the flow of trust information. ISO 22857¹ is an example of such a standard with respect to personal health data. It covers both the data protection principles that should apply to international transfers and the security policy which an organization should adopt to ensure compliance with those principles in collaboration with other organizations.
- In Section 3.5.1, we have introduced the aspects of e-contract for the enforcement of trust in B2B electronic commerce scenarios. We see that approaches such as the one investigated in [CCT03] and [KGV00] present a good basis for a methodology of e-contracts that enforces the policies on the privacy of the trust information from a high-level data representation and view down to the implementation layer.

4.4.1.3 Fulfillment of the requirements

Assuring the privacy of the data among members in online communities has never been easy. The contribution of the approach presented in this thesis, aims, in this regard, more at providing the involved parties with a logical method for tracking information that helps identifying violation of rules and negotiations in the CoT.

Although the concrete presentation of the realization of the discussed trust agreements shall be detailed in Chapter 5, Table 4.9 gives a first insight into the extent of the fulfillment of the requirements falling in these categories.

4.4.2 Policy Control

The extent to which policies shall be investigated with regard to trust management in this phase on the one hand refers to the privacy of the trust information. On the other hand, it refers to the way this trust information might be used for access control with regard to service usage and access to shared resources in the CoT. An example of

¹<http://www.iso.org/iso/>

Requirements	Fulfillment?
[ORG-TLA] [ORG-Time] [ORG-Integr] [ORG-Impact]	✓ In chapter 3, it has been demonstrated that most of these requirements are fulfilled within several approaches (see subsection 3.5.1).
[ORG-Simple] [ORG-Cost]	✓ By using standardized guidelines for describing the privacy policies in the CoT, a simple and a common understanding of the CoT operational rules can be assured. Moreover, the setup of the TLAs can be kept low.
[Priv-Collect]	✓ Setting the standardized guidelines imposes that any member in the CoT processing the trust information data, usually collected across domains, must comply with the privacy key principles about this data.
[Priv-Use]	✓ The same reasoning applies for the way the trust data may be handled and shared in the CoT.

Table 4.9: Fulfillment of the organizational requirements

the management of these access control policies is to deliberate a plan of conditions, followed by actions to guide decisions and achieve rational outcome.

For example, when a set of conditions is met then it can be ascertained that there is a potential solution for the requester to get access to the resource.

4.4.2.1 Privacy management aspects

From the perspective of privacy, the trust information can be seen as personal data of high value, since it is used for accomplishing collaboration as well as access decisions. Due to the fact that the privacy of this information plays an important role for the success or the failure of the CoT, it is obvious that a system facilitating the management and the exchange of this data must consider its aspects with a special care.

According to Clarke's definition [Cla99], privacy encompasses access control and a substantial degree of control over the use of personal data such that personal information can not be accessed by third parties, which have not been authorized for the use of that data.

With reference to that, we start with a basic way to model access control in a four-tuple (S, O, A, M) , where:

- S represents the set of subjects (the entities which access objects, the users resp. their agents).

- O represents the set of objects which are accessed by subjects.
- A represents the set of actions (access rights) performed on the objects (e.g. read, write, create).
- M represents the function that maps a tuple $(s, o, a) \in \{S \times O \times A\}$ to a decision $\in \{True, False\}$.

Usually, the mapping M can be stored in an access matrix, with rows corresponding to subjects, columns corresponding to objects, and matrix entries indicating the actions (access rights) that are allowed to the subjects. In practice, various access-control policies have been developed for storing and managing such an access matrix, which can be large and sparse, especially in distributed environments.

When we apply the given definitions to the privacy of the trust information, subjects correspond to the principals, the objects correspond to the requested information (trust levels, etc), and the actions correspond to the actions that may be performed on it.

In Subsection 3.3.1 we introduced two different classes of access control systems, (i) the discretionary access control systems such as ACL and (ii) the RBAC system, which represent an extension of ACL systems by assigning roles to subjects. However, due to the highly and dynamic changing nature of the trust relationships, we concluded from these discussions that none of these approaches can deal properly with the requirements on privacy and access control over the trust information, which shall be stored and updated dynamically in the trust matrix.

According to the definitions on privacy mentioned by Clarke in [Cla99], privacy management must consider two main issues, which can simply be expressed in the following two questions:

- What is the extent to which the information may be accessed?
- How can the information be handled by third parties?

In our trust assessment model, we distinguish between the trust information from past experience, which is collected automatically at the end of the interaction and that trust information by reputation, which is usually given manually by the users. Consequently, the rule-based approach for privacy management must consider access rights on both types of information.

For both types of trust information, we start from the principle that every CoT member receives write permission on his own data in the trust matrix, and read permissions for other members' data unless there are additional access and privacy rules that are against it. Note that these rules must be established formally in the trust agreements.

The overall processing of a write request for the trust information is depicted in Figure 4.22. We shall illustrate a concrete implementation example in Chapter 5.

So far we have discussed access control issues with regard to privacy management for the storage and the management of the trust information in the trust matrix. There, the subjects S involved in the access control diagram represent solely the CoT members, who may report about their experiences with other parties (either members or non-members of the CoT). The following subsection, however, addresses access control issues with respect to service usage and access to resources from unknown users.

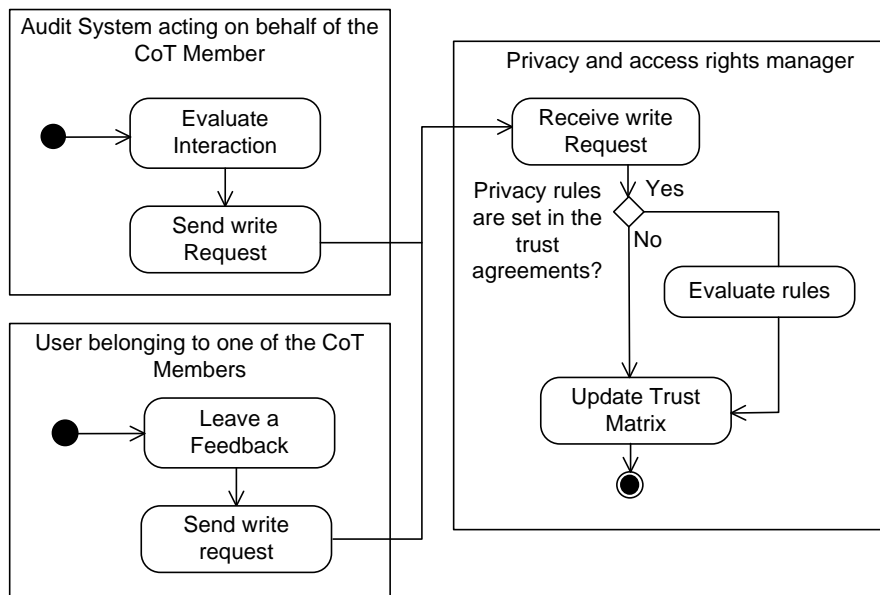


Figure 4.22: State diagram illustrating the processing of a write request of the trust information from past experience and by reputation

4.4.2.2 Access Control Management

In Chapter 2, we have discussed in three scenarios, that the administrative overhead for managing the ACLs in a dynamic scenario with a multitude of resources, users and services is increasing tremendously and difficult to keep controlled. Additionally, there is no support of unknown users as they can appear and leave in ad-hoc networks.

It has been concluded that a suitable approach to overcome these problems can be delivered by the trust-paradigm. That is, instead of or in addition to an inflexible ACL configuration, trust relationships between the members and their users can be utilized to gain access to the objects, so that it becomes possible to map features of natural trust between humans to the digital world.

Integrating trust management in the access control approach aims at including additional conditions along with the access control model; so that a requestor can be granted or denied access by checking his credentials as well as his trustworthiness represented in the so-called trust levels. Within that, we distinguish between automatic and static access control decisions:

- *Automatic access control decisions* provide the capability to meet the access decision automatically when it can be ascertained that the mandatory access conditions are fully met for the given access request. Based on these conditions, it can be ascertained, for example, when the requester belongs to a group of authorized users and his trust level is beyond a certain threshold, that the access can be automatically granted.
- *Static access control decisions*; these types of access controls rely heavily on human intervention, especially in situations where automatic access decisions cannot be set due to the lack of appealing features and information. Usually

static access control checks are all implemented via dynamic method calls, which forward the access decisions to the local administrators and resource owners.

4.4.2.3 Risk and trust based access decision workflow

A concrete delineation of access control decisions in relationship with trust and risk management parameters has been studied in previous work in [BH08]. It shall be presented in the following example, where the trust level is considered within the interval of $T_l \in [0, 1]$ and the risk levels in a quartile-based approach, resulting in levels *low*, *medium*, *high*, and *critical*.

Pseudo-code listing, presented in Algorithm 4.4.2.3 demonstrates the workflow for balancing of trust and risk exemplarily using the four risk levels defined above. Trust level thresholds of 0, 0.5, and 0.9 are used for access to resources with *low*, *medium*, and *high* risk respectively. In this example, decisions are delegated to an external policy decision point in two cases:

1. If the request is made by an external user which is yet unknown. This allows to handle anonymous access or self-enrolment on a per-service basis.
2. If the risk is *critical* and the user is fully trusted; this adds another layer of resource-local access control and can be used to combine traditional access control mechanisms with TBAC, which is a typical prerequisite in real-world scenarios.

In the current section, the theoretical part of the access control model of this work is provided, which apart from the given examples, may be applied to shared information spaces, offering the main benefit of gaining flexible and expressive access control without having to apply for a manual process.

However, from the technical realization point of view, the fulfillment of the interorganizational access control requirements as well as the technical realization requirements such as the queries, the policy management and the storage facilities shall be investigated in detail within the trust framework in Chapter 5.

4.5 Phase 4: Evolution

Obviously, the trust relationships estimated, stored and used for authorization purposes in the previous phases of the trust process model, should be monitored at each organization, as they might change in the course of time.

For each collaboration environment, a number of useful trust evaluation parameters can be defined. In Subsection 4.2.2.3 performance indicators as well as QoS parameters have been used for estimating trust from past experiences. The basic idea behind using these parameters for evaluating trustworthiness evolves around the frequency that the user violates a particular agreement and policy constraint. For example, a parameter for measuring the trustworthiness of a web service, for example, can be identified by the reliability of the service or the latency that data are returned by the service.

Algorithm 5 Exemplary trust and risk assessment

Input parameters: Subject s , action a , resource r , condition set C , subject's credential $Cred_s$, subject's action and resource specific trust level Tl_s , resource's risk level $\gamma_{(r,a,t_{now})}$

Output parameter: Access control decision, i. e. *permit* or *deny*

```

1: begin
2: if  $\exists s$  then
3:   return assessAccess( $Tl_s, \gamma_{(r,a,t_{now})}, C, Cred_s, a, r$ )
4: else
5:   // Set the default trust level for unknown users and delegate the decision
6:    $Tl_s := -1$ 
7:   return delegateDecision( $Tl_s, Cred_s, a, r$ )
8: end if
9: function assessAccess( $tl_s, \gamma_{(r,a,t_{now})}, C, Cred_s, a, r$ ):
10: access := deny // deny access by default
11: if ( $\forall c \in C : \mathbf{evaluateCondition}(c) == \mathbf{true}$ ) then
12:   if (
13:     ( $\gamma_{(r,a,t_{now})} == \mathit{low}$  and  $Tl_s \geq 0$ ) or
14:     ( $\gamma_{(r,a,t_{now})} == \mathit{medium}$  and  $Tl_s \geq 0.5$ ) or
15:     ( $\gamma_{(r,a,t_{now})} == \mathit{high}$  and  $Tl_s \geq 0.9$ ) ) then
16:     access := permit
17:   end if
18:   /* If the risk is critical, even fully trusted users may not access the resource
19:   without additional resource-local ruling */
20:   if ( $\gamma_{(r,a,t_{now})} == \mathit{critical}$  and ( $Tl_s == 1$ ) ) then
21:     return delegateDecision( $Tl_s, Cred_s, a, r$ )
22:   end if
23: end if
24: if access == deny then
25:   notify( $C, Cred_s$ ) // notify user about rejection reason
26: end if
27: log( $t_{now}, s, r, \mathbf{access}$ )
28: return access
29: end function
30: end

```

4.5.1 Monitoring

In this regard, we argue that anticipated monitoring tools on the one hand keep the trust information up-to-date, and on the other hand can significantly increase the quality of the entities' behavior in the CoT. That is, when they know before that the interactions in which they are involved are monitored, this helps their counterparts to anticipate potential untrustworthy behavior.

The given trust indicators represent the key feature for the monitoring phase of the life cycle of a trust relationship. Being regarded as characteristics or properties of an information system, they express the degree to which the partner can be expected to preserve the confidentiality, integrity, and availability of the information being processed, stored, or transmitted by the partner.

To realize such monitoring tools for continuously tracking changes to the trust information, the previously discussed trust indicators must be defined in a standard manner in the CoT. As we shall discuss in Chapter 5, the Resource Description Framework (RDF) represents the basic technology for representing the resources and the shared services in the CoT, where the properties can be used for linking resources with performance and quality parameters.

In addition to the RDF representation, these parameters must be referred to in the trust agreements as well. In these trust agreements, beside, (i) identifying common goals and objectives for the provision of services or information sharing in the CoT and (ii) agreeing upon the degree of trustworthiness needed to adequately mitigate the risk associated with the provision of such services or information sharing, the monitoring information based on the trust indicators must be agreed upon as well.

4.5.2 Assessment and evaluation of the monitoring information

Providing ongoing monitoring and oversight to ensure that the trust relationship is being maintained is decisive for the success of collaborations within a circle of trust. However, the evaluation of the monitoring information involves the following:

- Because the performance parameters defined in part by data communicated among the participants in the CoT (in the trust agreements) can be designed for a given set of service usage actions and resource consuming. In accordance with the present approach for assessing trust by means of these parameters, a change indicator need to be initialized at the start of a monitoring period.

Consequently, every organization in the CoT must evaluate the monitoring information according to the indicators agreed upon, and using the same evaluation rules.

- Audit tools are needed for extracting from an audit trail the information that is relevant for the trust assessment process. This utility can be directed to extract information for a particular user as well as for a service provider.

By means of the trust indicators and the shared rules, the audit tools enable to isolate a particular subset of data from a potentially large audit trail, and thus evaluate the status of the current interaction. Obviously the audit result can be either *successful* or *failed*.

4.6 Phase 5: Auditing and Change Management

As stated earlier, the monitoring of these parameters and the information resulting from the audit tools help determining if the collaboration continue to be trusted to operate within the agreed-upon rule. In the opposite case, the TBAC system must trigger a counter-measure automatically if some constraints are violated repeatedly.

Accordingly, enabling traceability of changes of the trust information leads to an additional process in the trust process model, the change management process, which takes these changes into consideration and performs the appropriate update operations.

The extent of the change, especially for the specification of the items the change should affect, has to be determined in the organizational agreements as well. In the following, only items that are relevant for the trust assessment shall be considered:

- *Trust level*; changes in the trust level of the principal might change the roles that the principal has in the CoT and thus the related privileges. As demonstrated in Phase 1, the trust levels can be assigned to the principal from different dimensions, where each level is related to a specific set of resources tied with a specific set of access privileges.

For this type of change, the update function demonstrated in Subsection 4.2.1.2 as well as the aggregation algorithm in Subsection 4.2.4 (in case there are changes from more than one dimension) can be applied. For example, when the audit tool indicates that a new interaction happened for a given scenario with a label "successful", the corresponding trust level for that scenario can be raised by means of the update function.

- *Trust agreements*; the second type of the change management regards changes in the trust agreements with respect to the trust indicators and the shared privacy and access policies, which are needed for the trust assessment process. During the monitoring period, if for example, a change is detected on the stored performance, the risk and QoS parameters, all related rules and algorithms must be changed and updated in response. The principals, who might be affected by these changes, must be notified as well.

Table 4.10 recalls the requirement on change management discussed in Chapter 2, and shows that the theoretical part of Phase 5 only ensures the fulfillment of alternative changes of the trust level, while the fulfillment of the remaining requirements shall be investigated in Chapter 5.

4.7 Evaluation and conclusion

In this chapter a trust process model is presented for CoT frameworks, in which principals with different aims and objectives are supposed to join and leave at any time. This process model has been developed and formalized in terms of different phases with different tasks in each phase.

- In phase 1, a logical method for extracting, estimating and aggregating trust from different dimensions has been developed within a set of algorithms and in partic-

Requirements	Fulfillment?
[Trust-Update] [Rep-Update]	✓ The fulfillment of these two requirements is assured by the update function as well as the aggregation algorithm.
[Sec-Update] [Risk-Update] [Notify]	? The fulfillment of these requirements shall be discussed in Chapter 5.

Table 4.10: Fulfillment of the change management requirements

ular mathematical formulas. An additional intermediary result up to here is also emphasized in an update function, which considers instantaneous changes that might influence the trust level and scales these changes in function of the amount of performed interactions by a principal.

- The main result of Phase 2 is the storage model proposed for helping the CoT members for storing and exchanging trust information from each member's own experience, since no member can know everything about its environment and no central authority can control all the collaborations.

Since the privacy of the stored information is of a great importance, the trust agreements established among the CoT members must thus address privacy protection. The establishment as well as the representation of the trust agreements in this context is one of the goals of the next chapter.

- Beside the management of the privacy policies of the trust information, which is subject of federation in the CoT, in Phase 3, an exemplary access control model is presented to show how trust and risk managements parameters can be mitigated for access control decisions with regard to service usage and resource consuming.
- Phase 4 identified the type of information that can serve as monitoring indicators and as a basis for the audit tools, whose goal is to evaluate the audit trails and estimate the status of the interaction, as successful or failed.
- In the same vein, Phase 5 distinguishes between two main change management processes. While change management, which regards the trust levels, can be solved in the previous phases by applying the update and the aggregation algorithms, the change management process regarding the resource description and the quality of services need to be handled within the trust agreements. This will be part of the focus of the next chapter.

Based on the concepts and issues presented in the different phases of the trust process model, the Trust Based Access Control (TBAC) Framework, which realizes these phases, shall be presented in the next chapter.

Chapter 5

Trust-Based Access Control Framework

"Trust only movement. Life happens at the level of events, not of words. Trust movement."

Alfred Adler

Contents

5.1	Conception of the TBAC Framework	185
5.2	Trust Broker	188
5.2.1	initializePackage	188
5.2.2	searchPackage	193
5.2.3	storagePackage	201
5.2.4	aggregatePackage	205
5.3	Storage Components	208
5.3.1	Trust Agreements Repository	210
5.3.2	Resource Description	215
5.3.3	Auditing the interactions	218
5.3.4	Identity Repository	220
5.4	Access Decision Engine (ADE)	221
5.4.1	Access decision policies	222
5.4.2	Privacy policies	223
5.5	Change Management	223
5.6	Summary and Conclusion	225

The realization of the trust process model, discussed in Chapter 4, represents the second part of this thesis. In this regard, the realization of this model is envisaged within a Trust-Based Access Control Framework (TBAC), which focuses mostly on the development of the mentioned methodologies and automated reasoning tools, including privacy and risk management aspects.

As it will be illustrated in the course of this chapter, the main feature of this framework is to be designed simple but at the same time extensible in order to define standard and

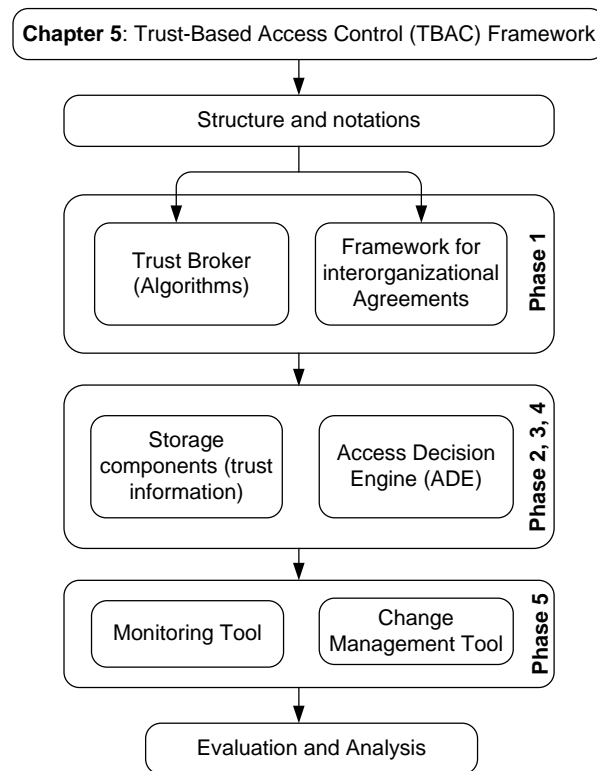


Figure 5.1: Sequence structure for Chapter 5

generic modules that enable interaction with a variety of sources and types of information in the FE. Following the order depicted in the sequence structure of this chapter, given in Figure 5.1, for the objective of modeling constructs for interorganizational trust and security, the TBAC Framework will encompass the following major components that carry specific responsibilities:

- **Trust Broker** represents an important element for managing the implementation of the different trust assessment algorithms, eliciting, evaluating and propagating the trust information in the CoT as described in Phase 1 of the trust process model in Section 4.2.
- **Trust Agreements Framework** is tightly coupled with the trust broker and is realized as a central component, where every member in the CoT (or the founder of the CoT on behalf of the members) can specify the trust requirements on the shared resources and services in the form of an agreement.

The specification of such collaboration requirements can be achieved according to the related agreements scheme and ontology in such a way that the performance and QoS parameters such as performance-value, cost of transaction, criticality of transaction, as well as several other different weights for direct and indirect transactions, and different weights for past and recent transactions can be expressed uniformly for the trust evaluation process.

- **Storage System**; the storage component of this framework has the role to contain the collection of trust information among the CoT members. Apart from data

structure, schemes and support for storage of this data, the privacy aspects on the usage of this information represent a major concern. In relation with the requirements specified in the *Agreements Framework*, the access rights on the trust information shall be specified accordingly.

- **Access Decision Engine (ADE)**; as the information being shared in the CoT will be classified by the information owner based on different legal and retention requirements, for example its value, sensitivity, consequences of loss or compromise, the information owner needs to determine the degree of verification (or trust) needed for users to perform transactions using that information, usually based on their history. The ADE component shall be realized to fulfill this aim.
- **Auditing and Change management tools**; this component shall be in charge of monitoring the collaboration process according to the requirements and the rules defined in the trust agreements. In this regard, trust values of service providers and consumers are evaluated and updated dynamically after the completion of each transaction.

The feature of automatic updates enables the consumer to receive the response from the broker significantly quicker compared to other reputation-based trust models where the trust values are computed at the request-time.

After giving a precise description of the notation used within each component of the TBAC Framework, the implementation aspects of each component shall be detailed in the following order: Section 5.2 specifies the package that constitutes the trust Broker. Section 5.3 illustrates the different types of storage repositories, and Section 5.4 addresses the access management policies within the ADE component.

In Section 5.5 a means for the specification of the change management process with regard to the life cycle of trust relationships in the CoT shall be described. Additionally, suitable measures for the realization of information change management according to the requirements discussed in Chapter 2 and Chapter 4 shall be presented. Finally Section 5.6 concludes this chapter with an evaluation on the basis of pertaining performance criteria.

5.1 Conception of the TBAC Framework

The components that constitute the design of the TBAC framework must conform to the functional building blocks discussed in previous chapters, because the design of the management system is highly dependent on the tools and utilities already deployed in the CoT, therefore, the grouping of the functions attributed to those building blocks within software components need not adhere strictly to a specific platform and must be generic.

As it can be seen in Figure 5.2, the implementation of the TBAC framework is built on top of the different components as introduced above, and the general principle for the realization of this framework is as follows:

When a respondent receives a request and wishes to rank the requester, he asks the trust broker to handle the request on his behalf. The trust broker executes the algorithms presented in Chapter 4 on a given CoT model and triggers an update of the trust matrix.

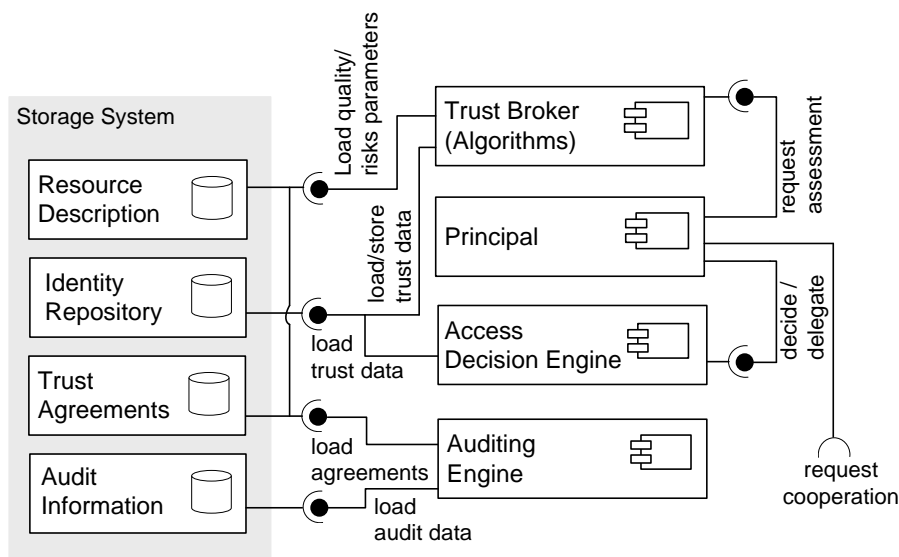


Figure 5.2: UML component diagram that represents the architectural components of the Trust-Based Access Control Framework

The trust matrix is represented by the identity repository of each CoT member (organization), where the identities of the principals (either users or organizations) including their credentials are managed, and can be made available to the trust broker. This latter represents the principals as nodes into a graph, where the weights of the connections between the nodes reflect the measure of trust in that relationship, which will usually be represented as a collection of attributes.

Based on the information provided by the Auditing Engine, which evaluates the interactions in contrast with the trust agreements and policies established in the Trust Agreements repository, the attributes defining new trust relationships shall be created, along with the identity of the principal, during the process of the interaction, as they tend to identify the other party as well as the level of trust in order to carefully estimate the probabilities of behavior in a given situation.

Finally, the component Access Decision Engine enables the respondent to decide on whether to grant access based on the results of the trust assessment workflows.

A more detailed representation of the management system can be depicted in the class diagram in Figure 5.3. This diagram illustrates a possible prototype implementation of the functional TBAC architecture. The correspondence between the components in the figure and their generic counterparts described in the course of this section is indicated by the light outline surrounding them. The numbered circles correspond to the interactions between the functional building blocks, which shall be described in the next sections.

While this section provides an overview of the components, and the classes chosen for the purpose of demonstration, it shall also discuss the selection of components for production use. A more detailed view on every building block in the focus of this work is given in Sections 5.2 through 5.3.

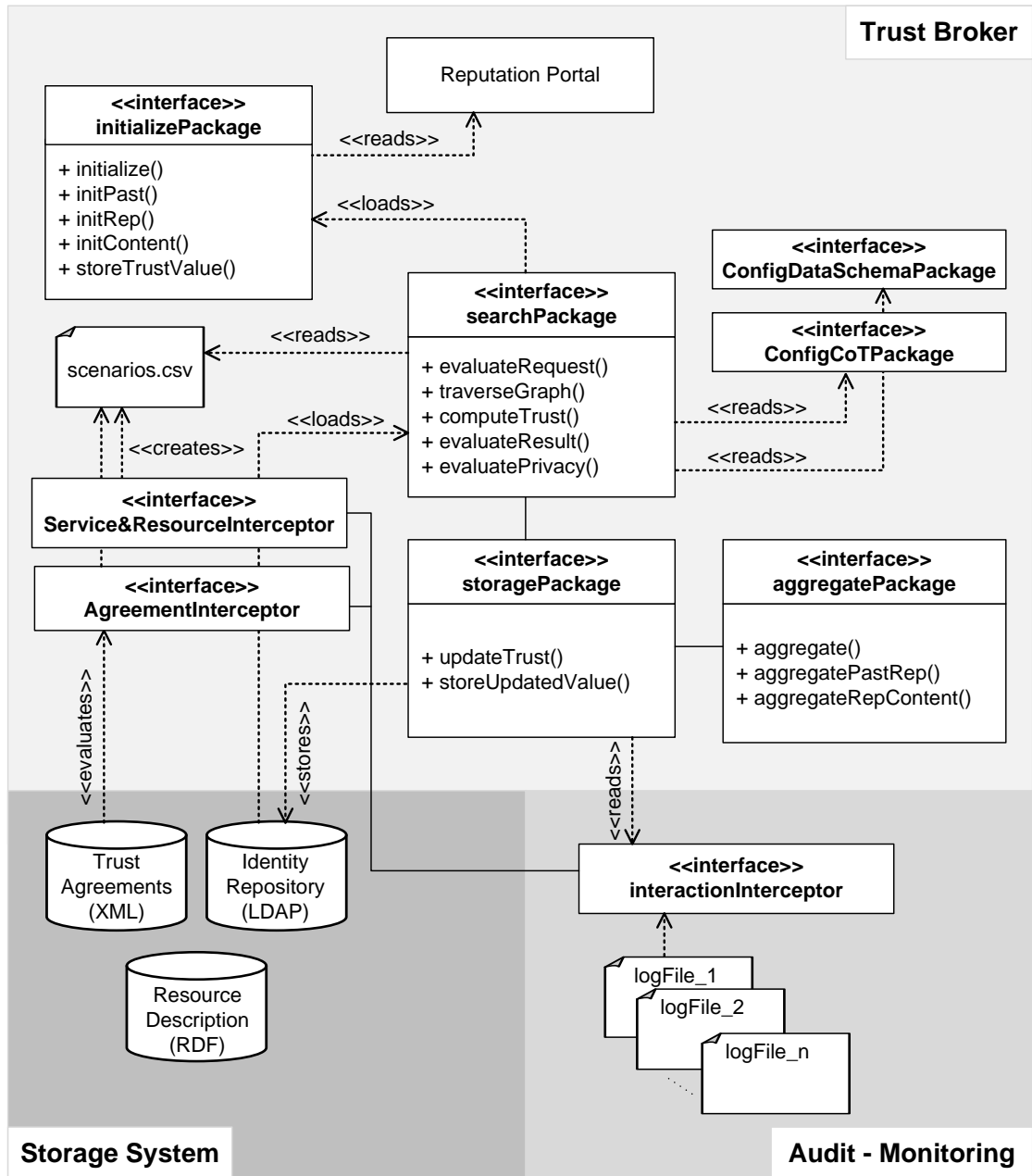


Figure 5.3: TBAC UML static class diagram

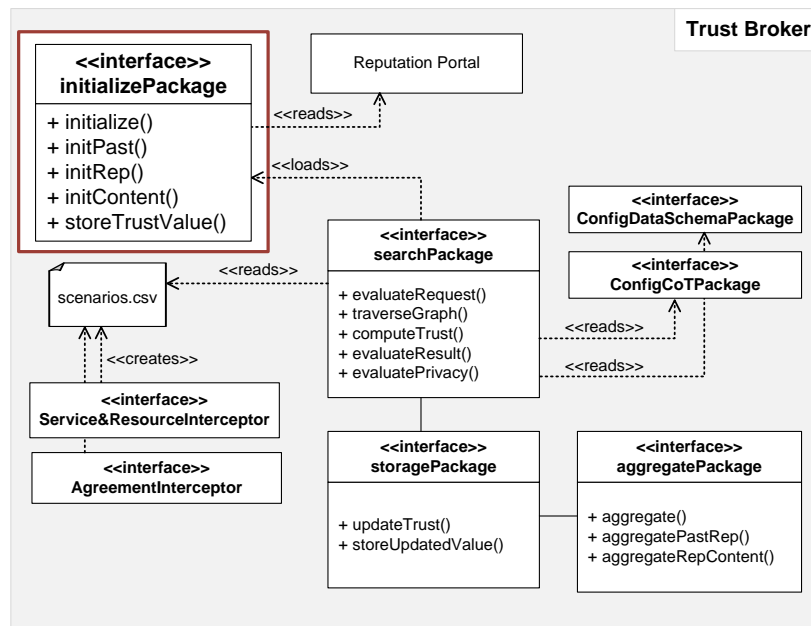


Figure 5.4: initializePackage

5.2 Trust Broker

Referring to Phase 1 of the trust process model, this component's objective is to first collect the relevant information about the requester, and then compute the prospective trust level by means of different input parameters. In this vein, the computation algorithms as well as the aggregation mechanisms can be applied in terms of mapping functions.

In the following, the detailed structure of the main packages that build the trust broker, including the function defined therein, shall be presented:

5.2.1 initializePackage

This package establishes the initialization steps that are required for the creation of the trust relationships among the members in the CoT. Among other principles of the initialization phase, it includes preparing and identifying the trust information structure, option keys, the location of the information as well as constructing access rules to this information over multiple providers.

In more detail, the functions encapsulated in this package, as illustrated in Figure 5.4, are defined as follows:

5.2.1.1 initPast()

The function `initPast()` is primarily based on Equation 4.4, presented in Subsection 4.2.2.3. This function investigates how principals can build trust exclusively based on their past experience, and ascertains that in doing so, automated trust building from

initPast ()		
Purpose		Read and prepare storage parameters for the trust values that are evaluated from the mechanism of <i>trust from past experience</i> .
Input Parameters	(P_1 , Org , P_x , $failedInter.$, $totalInter.$)	The parameters needed for the new trust relationship to be stored, where: <ul style="list-style-type: none"> - P_1 indicates the identifier of the first entity in the identity repository, which is requested for a given interaction. - Org indicates the parameter of the organization to which the requested entity belongs. - P_x indicates the identifier of the second entity involved in the interaction. - $failedInter.$ indicates performance information about failed interactions. - $totalInter.$ indicates performance information about all interactions performed among these two entities.
Returned Values	(P_1 , P_x , T_x , s)	Where: <ul style="list-style-type: none"> - T_x represents the trust level computed for the given interaction. - s indicates the information that represents the interaction in the form of a so-called trust scenario.
Related Functions	<code>pastStr ()</code>	This function provides the structure and the arrays <i>failedInter.</i> and <i>totalInter.</i>

Table 5.1: The function `initPast ()`

direct evaluation of peers' behavior during repeated interactions has a key role.

$$T_{l_{P_1 P_x}}(S : Res : Action : Param) = 100\% - \frac{\sum failedRequest}{\sum interaction/verification}$$

As detailed in Chapter 4, extracting trust from past experience can be enhanced by reasoning about the quality of the entities' behavior, and by taking the aspects of quality and performance parameters into consideration. In this respect, the function `initPast ()` initializes the data structure for representing the shared resources and services with these parameters, so that the ability to provide different priority to different applications, or data flows, or to guarantee a certain level of performance to a data flow can be efficiently verified. Table 5.1 illustrates the concrete parameters of this function.

The following enumeration type specifies the data structure used in the trust broker for associating resources with quality param-

eters. Note that these parameters are delivered by the modules `Service&ResourceInterceptor` and `AgreementInterceptor`, which shall be discussed in detail in Subsections 5.3.1 and 5.3.2.

```
$header1 = [
  ["resource", "action", "qualityParameter", "status"]
]
```

As shown above, while for resource representation, actions on the resources are associated with the quality parameters, the services instead may be associated directly to these parameters as follows:

```
$header2 = [
  ["service", "qualityParameter", "status"]
]
```

Based on this structure, the function `initPast()`, in combination with the module `interactionInterceptor` (this module shall be detailed in Subsection 5.3.3), assess the trust level as a percentage of the failed interactions from the total amount of interactions for a given scenario. In the following fragment listing of the function `initPast()` note that the values of 0/1 stored in the parameter "status" help to differentiate between failed interactions from successful interactions.

Listing 5.1: A code fragment of the function `initPast()`

```

1  ...
2  ...
3  ...
4  my $Failed = 0;
5
6  my $tableSizeF = $failedInteraction ->nofRow;
7  my $tableSizeT = $totalInteraction ->nofRow;
8
9  for(my $i=0; $i<$tableSizeT; $i++){
10     $res1 = $totalInteraction ->elm($i," resource ");
11     $act1 = $totalInteraction ->elm($i," action ");
12     $param1 = $totalInteraction ->elm($i," parameter ");
13
14     for(my $j=0; $j<$tableSizeF; $j++){
15         $res2 = $failedInteraction ->elm($j," resource ");
16         $act2 = $failedInteraction ->elm($j," action ");
17         $param2 = $failedInteraction ->elm($j," parameter ");
18         $status = $failedInteraction ->elm($j," status ");
19
20         if (($res1 == $res2)&&($act1==$act2)&&($param1==$param2)&&
21             ($status=="0")){
22             $Failed ++;
23         }
24     }
25 }
26 $Tx = 1-$Failed/$tableSizeT;
27 ...
28 ...
29 ...
```

initRep ()	
Purpose	Read and prepare storage parameters for the trust values that are evaluated from the mechanism of <i>trust by reputation</i> .
Returned Values	none
Input Parameters	(P_1 , Org , P_x , T_x , <i>indicator</i>) It uses the same parameters as in function <code>initPast()</code> , except for the parameter <i>indicator</i> which is used to map the reputation values to the trust scenario.
Related Functions	<code>reputationInterceptor</code>

Table 5.2: The function `initRep()`

5.2.1.2 `initContent()`

Following the same argumentation given for the function `initPast`, the function `initContent`, the objective of which is to assess trust about the content quality, relates the quality parameter directly to the resources instead of actions and subtypes of actions (see related discussions in Subsection 4.2.3).

In the same manner, this function assesses the trust level about the quality of a given resource as a percentage of the negative verifications when appraising the quality parameters from the total amount of resources that own the principal P_x .

$$T_{l_{P_1 P_x}}(Res : QualityParam) = 100\% - \frac{\sum failedVerifications}{\sum Resources(P_x)}$$

5.2.1.3 `initRep()`

This function's objective is to collect the reputation values, entered in the reputation portal (see Figure 5.3), and by means of the module `reputationInterceptor` adjusts all these values in relation to the performance and quality parameters that are referenced to as *scenarios*.

Analogical to the function `initPast()`, the function `initRep()` needs the module `Service&ResourceInterceptor` for associating the reputation values to the scenarios according to the standard resource and service description in the CoT. The parameters used by this function are given in Table 5.2.

5.2.1.4 `storeTrustValue()`

The resulting trust values from the previously defined mechanisms shall be stored by the function `storeTrustValue()`. Obviously, this information shall be used as a basis for the search package presented in Subsection 5.2.2, especially in situations when a request is received for future interactions thus reflecting similar scenarios.

By means of this function the trust information can be stored according to the properties of the following structure:

```
$header = ["voucher", "ID", "CoTMemberID", "Level",
           "Context", "Dimension"]
```

In the specified order, these properties indicate:

- (1) the identifier of the voucher node (the node that gives a trust value, thus starting a trust relationship),
- (2) the identifier of the connected node (the node getting a trust value),
- (3) the identifier of the home organization of the voucher node,
- (4) the corresponding trust level (this trust level may be extracted from the well-known trust assessment mechanisms),
- (5) the context of the trust relationship (the trust scenario), and
- (6) the dimension of the trust assessment method (past experience, content, or reputation).

Once the interaction has been performed and evaluated, the corresponding trust relationship can be stored in the identity repository by the principal denoted as a voucher. An exemplary realization of this storage operation is shown in Listing 5.2, where the identity repository is realized as an LDAP implementation. In such an implementation, the trust relationship can be attached as a sub-object beneath the object that represents the vouching node. As we shall discuss in Section 5.3, the schema that specifies the type of this sub-object is based on several attributes that basically reflect the same properties shown in the `$header` of the data structure of the trust relationships.

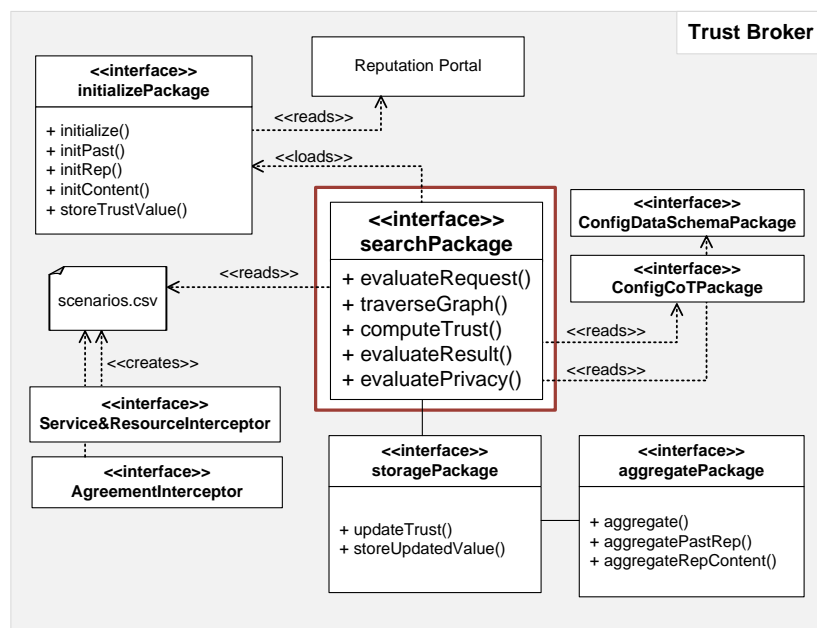
Beside the privacy rules that are managed by the module `AgreementInterceptor` (this module is detailed in subsection 5.3.1), each CoT member receives write permission on its own subcontainer in the directory tree, and read permissions for other members' subcontainers.

Listing 5.2: A code fragment of the function `storeTrustValue()`

```
1 $TR = $ldap->add('cn=trust($Px)', $trustSchema{"namingAttr"}=$P1, $dn'
2     attr => [
3         'cn' => [' $trustSchema{"namingAttr"}=$P1, $dn' ],
4         '$trustSchema{"memberID"}' => '$Px',
5         '$trustSchema{"context"}' => '$s',
6         '$trustSchema{"level"}' => '$Tx',
7         '$trustSchema{"dimension"}' => '$dimension',
8         'objectclass' => [' $trustSchema{"type"}'
9                             ... ],
10    ]
11 );
12 $TR->code && warn "failed to add entry: ", $TR->error ;
```

As discussed through the functions building the `initializePackage`, this package aims at preparing and archiving trust information from interactions taking place among known partners in the CoT. In coherence, the next package (`searchPackage`) is basically used to provide search properties and search content when a cooperation request is received from an unknown principal, or from a principal that requests a collaboration for the first time.

storeTrustValue()	
Purpose	Stores the trust information, collected from different assessment mechanisms, into the identity repositories.
Returned Values	none
Input Parameters	(P_1 , Org , P_x , T_x , s , $dimension$) See description in function <code>initPast()</code> .
Related Functions	<code>AgreementInterceptor</code>

Table 5.3: The function `storeTrustValue()`Figure 5.5: `searchPackage`

5.2.2 `searchPackage`

To achieve the search functionalities, this package uses various parameters to optimize and save searches, and provides access to several types of search functionality, each of which is described in detail in a set of functions, illustrated in Figure 5.5.

In general, most of the search functions work on the basis of the entered search criteria, which are combined with other information, such as the currently selected identity repository where the requested principal is located, as well as other parameters that deal with privacy agreements. The results of the search can then be displayed to the current requested node and written back in the corresponding trust information store.

In the following, the main functions building this package shall be described in detail:

evaluateRequest ()	
Purpose	Evaluates the request and prepare the input parameters to the following functions.
Returned Values	(P_1, Org, P_x, s) <p>Where:</p> <ul style="list-style-type: none"> - P_1 indicates the identifier of the requested entity in the identity repository. - Org indicates the parameter of the organization to which the requested entity belongs. - P_x indicates the identifier of the requester entity whose trust level is subject of verification and - s indicates the scenario of the interaction or the cooperation.
Input Parameters	none
Related Functions	<code>readInfo ()</code>

Table 5.4: The function `evaluateRequest ()`

5.2.2.1 `evaluateRequest ()`

As shown in Table 5.4, the main objective of this function in combination with the function `readInfo ()` is to receive requests and evaluate them in such a way as to determine how the search algorithm can best be assisted to address and improve dynamic search techniques.

Due to the fact that the input parameters delivered by this function can only be derived at the time of the request, placeholders in the identity repositories need to be denoted in a generic manner, indicating thus that potential trust information can be provided at runtime. Based on that, the identity repository must be configured in a flexible manner since fixed host-variable names, for example, in the distinguished names in the case of LDAP implementations cannot be defined beforehand.

Therefore, external configuration files, mainly `ConfigDataSchemaPackage` and `ConfigCoTPackage` in the class diagram presented in Figure 5.3, are used to enable ad-hoc extension of the input parameters for the schema design as well as the configuration parameters for identifying the providers in the CoT.

5.2.2.2 `traverseGraph ()`

The graph theory adopted by this function applies basic graph theory principles to sparse matrices, where the trust values are hold by each CoT member in the identity store. In this regard, each sparse matrix represents a graph based on the entries in the matrix, which in turn, represent the edges of the graph, and the values of these entries represent the associated weight (trust level) of the edge.

traverseGraph ()		
Purpose		Traverse the graph recursively by following adjacent nodes in a way that every edge and vertex in a graph can be visited in a systematic way. The graph will be optimized recursively with the function <code>computeTrust ()</code> .
Returned Values	T_x	This parameter represents the required trust level.
Input Parameters	(P_1, Org, P_x, s)	These parameters are defined in the function <code>evaluateRequest ()</code> .
Related Functions		<code>getEdge ()</code> , <code>computeTrust ()</code>

Table 5.5: The function `traverseGraph ()`

The function `traverseGraph(P, Px, Org, s)`, outlined in Table 5.5, traverses the graph G starting from the node indicated by the identifier $\$P1$ of the requested node whose identity store location can be specified by means of the parameter $\$Org$. Nonzero entries in the matrix indicate the presence of an edge. $\$principalT$ is a vector of target node indices that are connected to the requester node $\$Px$. They are listed in the order in which they are discovered and are represented according to the same properties used in function `storeTrustValue ()`:

```
$header = ["voucher", "ID", "CoTMemberID", "Level",
           "Context", "Dimension"]
```

Note that the search break conditions of this algorithm are established by means of the following statement:

```
unless ($indicator ne $P1 || undef($principals))
```

Expressing thus two main conditions:

- Unless the node index `$indicator` returned by the recursive call finds a path between the original node $\$P1$ to the target node $\$Px$, or
- there are no more neighboring nodes (`$principals`) of $\$P1$, which might be asked for eventual relationship to $\$Px$.

In Listing 5.3, a more detailed code fragment of the function `traverseGraph ()` is illustrated.

Listing 5.3: A code fragment of the function `traverseGraph ()`

```
1 ($Tx, $principals, $principalT) = getEdge($P, $Px, $Org, $s);
2 if (Tx eq "-1"){
3   unless ($indicator ne $P1 || undef($principals)){
4     if (($principalT->nofRow) = 0){
5       $principalMiddleS = $principals;
6
7       unless ((( $finalprincipalT->nofRow) != 0) ||
8               undef($principalMiddleS)){
9         $stableSize = $principalS->nofRow;
```

```

10
11     ### Ask the neighbors one level below
12     unless ($i >= $stableSize){
13         $ID = $principalS ->elm($i,"ID");
14         $MemberID = $principalS ->elm($i,"CoTMemberID");
15         ($TxMiddle, $principalMiddleS, $principalMiddleT) =
16         getEdge($ID, $Px, $MemberID, $s);
17
18         if(($principalMiddleT ->nofRow) != 0){
19             $OrgPx = $principalMiddleT ->elm(0,"CoTMemberID");
20             $PxLevel = $principalMiddleT ->elm(0,"Level");
21             $finalPrincipalT ->addRow([ $ID, $Px, $OrgPx,
22             $PxLevel, $s], ($principalT ->nofRow)++);
23         }
24         ### Collect principalMiddleS
25         $finalprincipalS += $principalMiddleS ->clone();
26
27         $i++; ### visit the next neighbour located in the same
28             level
29     }
30     ...
31     ...
32     ...
33 }

```

Listing 5.4 provides additional arguments for the search principle of this function and shows that in the absence of potential nodes (`$principalT`) relating the requested node `$P` to the requester node `$Px`, `$principalS` is then returned to continue the search one level further down in the graph. These nodes represent a vector of neighboring node indices (listed in the order of the node indices) that are connected level by level to the original node `$P`.

Listing 5.4: A code fragment of the function `traverseGraph()`

```

1     ...
2     ...
3     ...
4     ### Compute the trust level
5     if(($finalprincipalT ->nofRow) != 0) {
6         ($Tx, $indicator) = compute($finalPrincipalT, $principalS);
7     } else {
8         for(my $j=0; $j<$principalMiddleS ->nofRow; $j++){
9             $id= $principalS ->elm($j,"ID");
10            $memberID = $principalS ->elm($j,"CoTMemberID");
11            traverseGraph($id, $Px, $memberID, $s);
12            $PrincipalS ->addRow([ $ID, $Px, $OrgPx, $PxLevel, $s],
13            ($principalS ->nofRow)++);
14        }
15        ...
16        ...
17    }

```

By means of the statement described below, it can be ascertained that once all the nodes one level lower have been visited, the function `computeTrust()` shall be called step by step in order to optimize the path between the nodes whose edges are stored in `$principalS` and those that are located one level beneath and stored in `$finalPrincipalT` as they might be connected to the requested node P_x .

```

1 if(($finalprincipalT ->nofRow) != 0) {
2     ($Tx, $indicator) = compute($finalPrincipalT, $principalS);
3 }

```


computeTrust ()	
Purpose	Compute the trust level on the basis of the weight of the edges connecting the requester and the requested node. The weight of these edges are provided by the function <code>traverseGraph()</code> . The implementation of this function is based on Algorithm 2, presented in Section 4.2.2.4
Returned Values	<p>(P, T_x)</p> <ul style="list-style-type: none"> - T_x indicates the trust level, which might be computed for a portion of the graph, as this function performs a breadth-first search. - P indicates the level in the graph in which the computation took place.
Input Parameters	<p>$(@principalT, @principalS)$</p> <ul style="list-style-type: none"> - $@principalT$ represents the array of the nodes whose edges are directly related to the requester node. - $@principalS$ represents the array of the nodes whose edges are directly related to the requested node.
Related Functions	none

Table 5.6: The function `computeTrust ()`

5.2.2.3 `computeTrust ()`

In coherence with the function `traverseGraph()`, the function `computeTrust ()` simply employs a filtering algorithm that computes the overall path between the edges that compose the graph into different levels.

Exactly as it has been demonstrated in Algorithm 2 in Subsection 4.2.2.4, this function optimizes, little by little, the path between every two levels in the graph for improving the accuracy of the optimization. As shown in Listing 5.5 the edges of these levels are stored in the arrays `$principals` and `$principalT`.

getEdge ()	
Purpose	Search for the weight of the edges (trust level) in the identity information stored in the sparse matrices. It might return an unknown status if the desired edge cannot be found, because this algorithm looks only at the connectivity described by the sparse matrix by each CoT member.
Returned Values	<p>$(T_x,$ $principalS,$ $principalT)$</p> <ul style="list-style-type: none"> - T_x indicates the trust level, either found from neighboring nodes or unknown (see listing 5.6). - $@principalT$ represents the array of the nodes whose edges are directly related to the requester node. - $@principalS$ represents the array of the nodes whose edges are directly related to the requested node.
Input Parameters	<p>$(P_1,$ $Org,$ $P_x, s)$ These parameters are defined in the function <code>evaluateRequest ()</code>.</p>
Related Functions	none

Table 5.7: The function `getEdge()`Listing 5.5: A code fragment of the function `computeTrust ()`

```

1  ...
2  for(my $i=0; $i<($principalS->nofRow); $i++){
3
4      my $P = $principalS->elm($i,"voucher");
5      my $ID = $principalS->elm($i,"ID");
6
7      for(my $j=0; $j<($principalT->nofRow); $j++){
8          if($principalT->elm($j,"voucher")==$ID){
9
10             if($principalT->elm($j,"level") < $principalS->elm($i,"level")){
11                 $M += $principalT->elm($j,"level") * $principalS->elm($i,"level");
12             }else{
13                 $M += $principalS->elm($i,"level") * $principalS->elm($i,"level");
14             }
15         }
16     }
17     $N += $principalS->elm($i,"level") * $principalS->elm($i,"level");
18 }
19 $Tx = $M/$N;

```

5.2.2.4 `getEdge ()`

Beside the function `computeTrust ()`, the function `traverseGraph ()` depends on the function `getEdge ()` to test whether or not an edge between two nodes exists, and subsequently retrieves a reference to the edge between the specified nodes.

In Listing 5.6, based on an LDAP implementation, it can be seen that the representation of the edges does not need to allocate a static memory area for that purpose, since most graphs created are sparse and have a small number of nodes.

Listing 5.6: A code section of the function `getEdge()`

```

1 foreach my $entry ($search->entries) {
2
3     $ID = $entry->get_value('trustSchema{"ID"}');
4     $MemberID = $entry->get_value('trustSchema{"MemberID"}');
5     $T = $entry->get_value('trustSchema{"level"}');
6     if(($entry->get_value('trustSchema{"ID"}') eq $Px)){
7         if($P == $P1){
8             print "Direct Edge to the requester is found!";
9             $Tx = $entry->get_value('trustSchema{"level"}');
10        }else{
11            ### One of the neighbors has a link to the requester
12            $principalT->addRow([$P, $ID, $MemberID, $T, $s], $counter);
13        }
14    }
15    else{
16        ### none of the neighbors has a link to the requester
17        $principalS->addRow([$P, $ID, $MemberID, $T, $s], $counter);
18    }
19    $counter++;
20 }

```

5.2.2.5 evaluateResult()

The principle of the evaluation of the returned values from the function `traverseGraph()` is handled by the function `evaluateResult()` (see Table 5.8 for details about its parameters as well as related functions).

As demonstrated in Listing 5.7, in the absence of the trust level for the given scenario, this function attempts to search the trust level of the same principal but for other alternative scenarios. These shall be then returned as an array `$trustValues`.

Listing 5.7: A code section of the function `evaluateResults()`

```

1
2 if($Tx eq "-1"){ ### Trust Level not found for the requested scenario
3     ### Search the trust level for other alternative scenarios
4     @scenarios = search_scenario();
5     if(@scenarios){
6         foreach my $i(@scenarios){
7             $T = traverseGraph($P1, $Org, $Px, $i);
8             $trustValues->addRow([$T, $s], ($trustValues->nofRow)+);
9         }
10    }
11 }
12     if(Tx ne "-1"){ ### Trust Level found for the requested scenario
13     return $Tx;
14 }else{
15     return $trustValues;
16 }

```

5.2.2.6 searchScenario()

As introduced in function `evaluateResult()`, the function `searchScenario()` extracts the possible trust scenarios and thus enables all

evaluateResult ()	
Purpose	Evaluate the resulting trust level and search for additional trust values related to other scenarios when the requested scenario is missing.
Returned Values	<ul style="list-style-type: none"> - T_x indicates the requested trust level for the given scenario. - <code>@trustValues</code> indicates the trust level for other alternative scenarios in case the requested scenario is missing. Note that this function returns -1 for unknown trust level.
Input Parameters	(P_1, Org, P_x, s, T_x) In addition to the parameters defined in the function <code>evaluateRequest()</code> , T_x indicates the computed trust level, returned from the function <code>traverseGraph()</code> .
Related functions	<code>searchScenario()</code> , <code>traverseGraph()</code>

Table 5.8: The function `evaluateResult()`

searchScenario ()	
Purpose	Load all possible trust scenarios.
Returned Values	<code>@scenarios</code> indicates the alternative scenarios that are extracted from the QoS about the shared resources and services
Input Parameters	none
Related functions	none

Table 5.9: The function `searchScenario()`

the CoT Members to match advertised quality levels for their services and resources with QoS preferences.

These preferences are referenced in a common file `scenarios.csv` (see the listing below), which can be read by every CoT Member, since it can be generated by means of the modules `Service&ResourceInterceptor` and `AgreementInterceptor`. Each call of these modules generates this file automatically thus reflecting the most current status of the scenarios entered in the Agreements as well as in the Resource Description Repository. This component shall be described in detail in Section 5.3.

```

1 $content = $f->load_file('scenarios.csv');
2 @scenarios = split /\;/, $content;

```

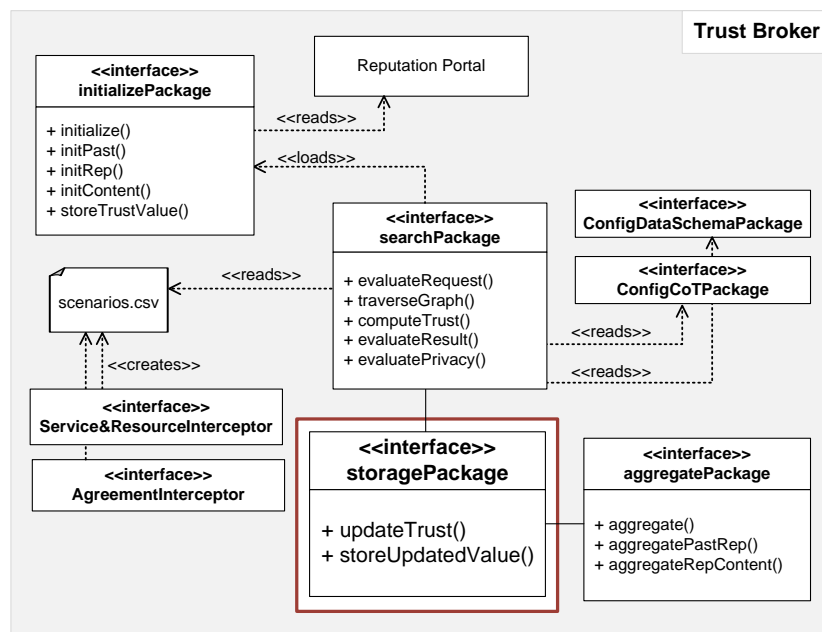


Figure 5.6: storagePackage

5.2.3 storagePackage

While the previous two packages address issues related to the initialization and the assessment of the trust levels, especially in the absence of direct relationship to the collaboration partner, instead, the `storagePackage` is intended to ensure functionalities for keeping the trust information and the trust relationships in the CoT up-to-date over time.

In this respect, when the management aspects of the trust information are in place, and the information transport facility among the CoT members has been successfully configured, the operational functions of this package can be placed in operation. Figure 5.6 gives an overview of the main functions that influence the execution of this package.

5.2.3.1 updateTrust ()

This function can be applied to cover two different scenarios:

1. When an interaction is performed for the first time, this function considers the creation of a new node (a new sub-object in LDAP terms) by inserting a node in the specified position in the graph and consequently adding a new edge, which reflects the corresponding relationship.

In this case, the implementation of the update function takes the same shape as the function `storeTrustValue()` defined in the `initializePackage`.

2. When the interaction is performed with a principal node, which already exists in the identity repository, the function `updateTrust()` will update the nodes relationships according to the evaluated quality of the performed interaction so

updateTrust ()	
Purpose	Update the trust level from different update mechanisms.
Returned Values	none
Input Parameters	(P_1 , Org , P_x , $dimension$, $totalInteraction$)
Related Functions	<ul style="list-style-type: none"> - <code>traverseGraph()</code> is needed for reading the initial trust value that shall be updated. - <code>storeUpdatedValue()</code> is needed for storing the updated trust value in its original position.

Table 5.10: The function `updateTrust ()`

that the relationships will be relocated in order to fit the new one at its specified position.

Here, the update function introduced in Subsection 4.2.1.2 in Chapter 4 shall be applied for increasing or decreasing the existing trust level according to the changes collected either from the auditing module `interactionInterceptor` or the `Reputation Portal` in the class diagram presented in Figure 5.3.

The update formula given in Subsection 4.2.1.2 updates the trust level according to the statement of $T_l(t) = T_l(t - 1) \pm \Delta T_l$, where ΔT_l is influenced by $interaction(\chi)$ as follows:

$$\Delta T_l = \begin{pmatrix} 1 - \frac{1}{2}e^{-\alpha(\sum interaction(\chi))} & \text{if } 0.5 < \chi < 1 \\ 0 & \text{if } \chi = 0.5 \\ \frac{1}{2}e^{-\alpha(\sum interaction(\chi))} & \text{if } 0 < \chi < 0.5 \end{pmatrix}$$

In relation with the mechanism used for assigning the changes of the trust level, the variable χ may have the following values:

- **Trust from past experiences:** The module `interaction-Interceptor` may provide the values of 0/1 distinguishing thus between `Failed` interactions and `Successful` interactions.
- **Trust by reputation:** The values entered in the `Reputation Portal` may have the following values: $\chi = 1$ for a positive rating, $\chi = 0.5$ for a neutral rating, and $\chi = 0$ for a negative rating.

As illustrated in Table 5.10, the implementation of this function very much depends on the input parameters *dimension*, which indicates the mechanism used for allocating the variable *delta*. This latter indicates the content of the change χ that is introduced in the update function.

By using exactly the same data structure introduced in the `initializePackage` (see below) for representing the shared services and resources, the variable *delta* can be extracted from the parameter "status". Listings 5.8 and 5.9 show the concrete usage of these parameters in both cases.

```
$header1 = [
  ["resource", "action", "qualityParameter", "status"]
]
```

Listing 5.8: A code fragment of the function `updateTrust()` for updating the trust values from past experience

```

1 ...
2
3 $Tx = traverseGraph($P1, $Px, $Org, $s);
4
5 for(my $j=0; $j<($totalInteraction->nofRow); $j++) {
6
7   my $res      = $totalInteraction->elm($j,"resource");
8   my $act      = $totalInteraction->elm($j,"action");
9   my $param    = $totalInteraction->elm($j,"parameter");
10  my $status    = $totalInteraction->elm($j,"status");
11  my $dimension = $totalInteraction->elm($j,"dimension");
12
13  if($status=="0")
14    $failed ++;
15  else
16    $success ++;
17 }
18
19 if($dimension eq "past") {
20   if($failed){
21     $delta = 0.5*exp($alpha * $failed);
22     $Tx -= $delta;
23   }
24   if($success) {
25     $delta = 1-0.5*exp($alpha * $success);
26     $Tx += $delta;
27   }
28 }
29 ...
```

Note that the parameter `$alpha` is the convergence factor of the update function that controls the increment or the decrement of the trust level in function of the amount of interactions.

storeUpdatedValue ()	
Purpose	Store the updated trust levels back into the identity repositories.
Returned Values	none
Input Parameters	(P_1, Org, P_x, s, Tx)
Related Functions	evaluatePrivacy ()

Table 5.11: The function storeUpdatedValue ()

Listing 5.9: A code fragment of the function updateTrust () for updating the trust values from the reputation values

```

1
2 ...
3 if ($dimension eq "rep"){
4     if ($repValue == "0"){
5         $delta = 0.5*exp($alpha * $failed);
6         $Tx -= $delta;
7     }
8     ### Do nothing if ($status == "0.5")
9     if ($repValue == "1"){
10        $delta = 1-0.5*exp($alpha * $success);
11        $Tx += $delta;
12    }
13 }
14 ...

```

5.2.3.2 storeUpdatedValue ()

Obviously, any update that is triggered by the audit component or the reputation portal has to be saved in order to be used again later. This feature allows frequently repeated interactions to be evaluated and stored so that the trust information remains consistent. In doing so, it ensures the durability of the lifecycle of trust relationships among the partners in the CoT as well.

Listing 5.10: A code fragment of the function storeUpdatedValue ()

```

1
2 $TR = $ldap->modify('cn="trust($Px)", $trustSchema{"namingAttr"}=$P1, $dn',
3     add => {
4         ...
5         $trustSchema{"level"}' => '$Tx',
6         ...
7     }

```

An example implementation of this update can be demonstrated in Listing 5.10. Similar to the function storeTrustValue () from the initializePackage, the update by means of the LDAP protocol modifies the content of the entry given by the distinguished name DN on the server, where the entity that evaluated the interaction is located.

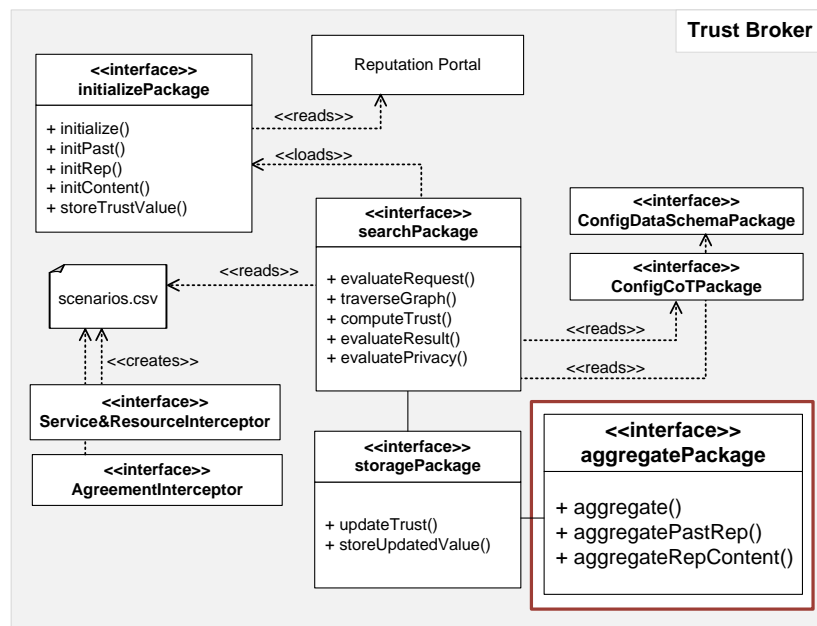


Figure 5.7: aggregatePackage

5.2.3.3 evaluatePrivacy()

All issues concerning the privacy constraints and rules for the management of the read, browse and write permissions on the trust information are discussed in Subsection 5.4.1.

5.2.4 aggregatePackage

The aggregation package considers the problem of combining several trust assessment results from various computation methods and information sources.

As it has been discussed in the previous packages, the trust values may be generated from techniques of *trust from past experiences* regarding both behavior and content trust, as well as *trust by reputation*. That is, the need of aggregating several alternatives based on one or more criteria is encountered very often in collaborative environments.

The main functions building this package are based on the aggregation algorithm, discussed in Subsection 4.2.4 in Chapter 4, which includes combining evaluation functions, selecting information documents based on multiple criteria, and improving the precision of the trust evaluation algorithm through description associations.

In the following, each aggregation method shall be discussed in the same order as depicted in Figure 5.7.

aggregate ()	
Purpose	Aggregate trust values that are provided from different assessment methods.
Returned Values	none
Input Parameters	<i>Trust</i> This variable, which follows the same data structure ["voucher", "ID", "CoTMemberID", "Level", "Context", "Dimension"] provides the relevant parameters of the new trust values.
Related Functions	traverseGraph(), aggregateTwoDimensions(), storeUpdatedValue()

Table 5.12: The function `aggregate ()`

5.2.4.1 `aggregate ()`

This function can be regarded as an extension of the function `updateTrust ()`. While this latter updates existing trust values with new ones when they originate from the same mechanism, instead, the function `aggregate ()` performs additional aggregation techniques when the trust values address the same context (the trust scenario), but are generated from different assessment methods.

By means of the function `traverseGraph ()`, as illustrated in Listing 5.11, it can be verified whether a trust level from the same dimension and for the same context exists. If so, the same principle of the update function shall be applied on the variable *TxOld*.

In the absence of a previous trust level (line 28), this variable shall be replaced by the variable *TxNew*, which is provided as an input parameter.

Listing 5.11: A code fragment of the function `aggregate ()`

```

1
2 sub aggregate{
3   my $Trust = $_;
4
5   my $P1      = $Trust->elm(0,"voucher");
6   my $Px      = $Trust->elm(0,"ID");
7   my $Org     = $Trust->elm(0,"CoTMemberID");
8   my $TxNew   = $Trust->elm(0,"Level");
9   my $s       = $Trust->elm(0,"Context");
10  my $dimNew  = $Trust->elm(0,"Dimension");
11
12  ($TxOld, $dimOld) = traverseGraph($P1, $Px, $Org, $s);
13
14  if (!$TxOld){
15    if ($dimNew eq $dimNew){
16      ### Perform an update
17      if ($TxNew < 0.5){
18        $delta = 0.5*exp($alpha*$TxNew);
19        $TxOld -= $delta;
20      }
21      if ($TxNew >= 0.5){
22        $delta = 1-0.5*exp($alpha*$TxNew);
23        $TxOld += $delta;
24      }
25    }

```

```

26 ...
27 ...
28 }else{
29   ### There are no previous trust levels for the required scenario
30   $TxOld = $TxNew;
31 }
32 storeUpdatedValue($P1$, $Org$, $P_x$, $s$, $TxOld);

```

Listing 5.12 deals with the case where the new and the old trust values referring to the same context are generated from different trust dimensions. As already discussed in Section 4.2.4, when one of the two values is generated from past experience and the second from trust by reputation, this approach considers the first value as the starting value and increments it or decrements with the value assigned from the `Reputation Portal` according to function `updateTrust`.

The reason for that is that after each interaction in the CoT, computing the trust level from past experience is expected to be automatically performed by means of the monitoring tools, while it cannot be ensured beforehand that the interaction partner leaves a rating level at the end of the interaction.

The same reasoning applies for the aggregation between the content trust and trust by reputation.

Finally the aggregated trust level shall be stored back in its original position in the corresponding identity repository (line 32 in Listing 5.11).

Listing 5.12: A code fragment of the function `aggregate()`

```

1 ...
2 ### In the case the trust level is available from more than one dimension
3
4   ### Aggregation between Trust by Reputation and Trust from Past Experience
5   if(($dimOld eq "past") && ($dimNew eq "rep")){
6     $TxOld = aggregateTwoDimensions($TxOld, $TxNew);
7   }
8   if(($dimOld eq "rep") && ($dimNew eq "past")){
9     $TxOld = aggregateTwoDimensions($TxNew, $TxOld);
10  }
11
12  ### Aggregation between Trust by Reputation and Trust Content
13  if(($dimOld eq "content") && ($dimNew eq "rep")){
14    $TxOld = aggregateTwoDimensions($TxOld, $TxNew);
15  }
16  if(($dimOld eq "rep") && ($dimNew eq "content")){
17    $TxOld = aggregateTwoDimensions($TxNew, $TxOld);
18  }
19  ...

```

5.2.4.2 `aggregateTwoDimensions()`

As stated earlier, this function applies the principles of the function `updateTrust()` for aggregating two trust levels. Note that the order in which the input parameters are given to this function is very important, since the first value shall be modified in function of the values of the second one.

aggregateTwoDimensions ()		
Purpose		Aggregate the trust level issued from two different assessment mechanisms.
Returned Values	$T1$	It represents the final modified trust level.
Input Parameters	$(T1, T2)$	These two parameters represent the trust levels that shall be aggregated into one final value.
Related Functions	none	

Table 5.13: The function `aggregateTwoDimensions ()`

The case where the trust level is available from more than one dimension.

Listing 5.13: A code fragment of the function `aggregateTwoDimensions ()`

```

1
2 sub aggregateTwoDimensions{
3   my ($T1, $T2) = @_;
4   ### Perform an update
5   if($T2 < 0.5){
6     $delta = 0.5*exp($alpha*$TxNew);
7     $T1 -= $delta;
8   }
9   if($T2 >= 0.5){
10    $delta = 1-0.5*exp($alpha*$TxNew);
11    $T1 += $delta;
12  }
13  return $T1;

```

Due to the heterogeneity of the trust information in FE, the aggregation package may encompass a wide set of functions and techniques. In this section, some particular use cases of an aggregation algorithm that collates dissimilar responses have been studied in detail. We argue that these examples provide a good basis on which additional aggregation functions can be investigated.

The implementation of the aggregation package represents the last step for the realization of the trust broker as shown in the class diagram in Figure 5.3.

In the following sections, the implementation as well as the evaluation of the remaining building components of the TBAC Framework shall be detailed in the successive order outlined in Section 5.1.

5.3 Storage Components

The analysis of the trust broker in the previous section has shown that this component is tightly coupled with the storage component, because most of the challenging aspects for assessing trust evolve around initializing, searching, evaluating, and updating the trust information in the CoT.

In this subsection, we are not proposing to investigate new storage technologies. However, we are taking existing approaches in the design of the storage component, and in particular, we are adopting the viewpoint that this component can be constructed with other sub-components that scale up and are suitable for the multiple types of the information in the CoT.

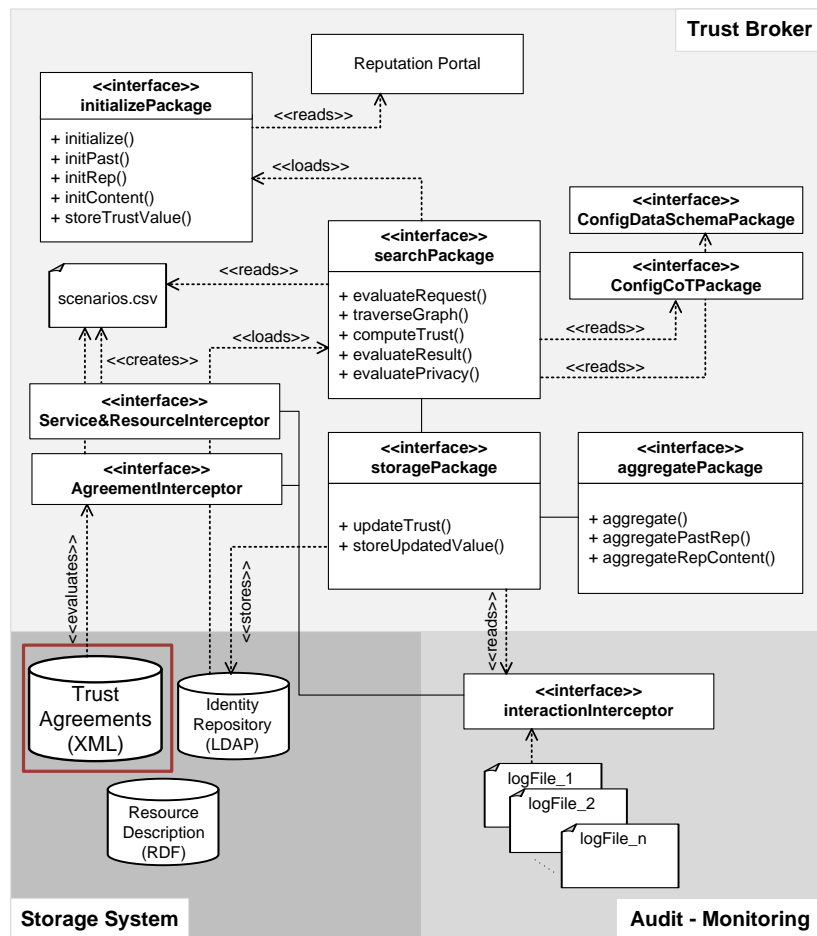


Figure 5.8: Trust Agreements Repository

Briefly, the storage component comprises the following sub-components:

- **Trust Agreements Repository** represents the different Service Level Agreements as well as any other operational agreements established among the members in the CoT.
- **Resource Description Repository** In combination with the Trust Agreements Repository, this component represents the services and resources for inter-domain access management (including the QoS and performance parameters).
- **Identity Repository** manages the storage of the identity information of the principals in the CoT.
- **Audit Information Repository** represents relevant information for evaluating the quality of interactions.

5.3.1 Trust Agreements Repository

Trust agreements and trust declarations are some of the most common collaboration aspects being required in several application fields. Like any other contracts, trust agreements need to be setup in the system and given their emplacement accessible to make them traceable and to ensure they are administered properly.

According to the definitions given in Chapter 2 in Subsection 2.1.2.6, the aspect of central management of the trust agreements in several models of the CoT has proven to be very efficient to minimize administrative costs and to ensure successful running of the CoT.

However, trust relationships are a complex type of agreements and this work has only highlighted a few key points. Moreover, the preparation of a trust agreement or declarations of trust in most collaborative and business scenarios refer to the legal counsel. Therefore, the purpose of this study is merely to provide a methodology that allows to generate an automatic evaluation of existing trust agreements, and thus, help the involved parties, as represented by their unions, to test the behavior of each other based on the policies included within the scope of these trust agreements.

Figure 5.8 recalls the class diagram of the TBAC Framework, presented in Section 5.1, and shows how the Trust Agreements Repository interacts with the other components in the different building blocks.

5.3.1.1 QoS ontology

For the purpose of our study, many ontology and eContract languages exist in the literature. Due to the fact, that except for trust by reputation, all the other dimensions of trust assessment are mainly based on quality and performance parameters, we argue that the ontology as well as the QoS policy proposed by Michael Maximilien et al. in [MS04a], [MS04b] and [Max05] for dynamic selection of web services can serve as a good starting point to realize this sub-component.

The QoS specification in this approach is realized by means of an ontology that allows to match services semantically and dynamically. By using the same ontology, the providers have the possibility to express their policies, on the one hand, and the consumers express their preferences on the other hand. This represents a key feature for implementing the component `AgreementInterceptor` by using the provider's advertised QoS policy for the services and the consumers' QoS preferences to monitor the behavior and record consumer and service interactions.

While the service ontology relates services to QoS, the QoS ontology deepens the quality concepts. In the following the main concepts of this ontology are outlined briefly:

- **Quality:** Represents a measurable nonfunctional aspect of the service. Quality instances have measurable attributes, which have relationships with each other.
- **QAttribute:** Represents the set of attributes that constitute the value as well as the type of the Quality concept.
- **QMeasurement:** Represents the measurement method of the concept of quality. This measurement can be objective (made automatically via a software agent) or subjective (made via some human agent).

- **QRelationship:** Represents the relationships among the qualities into main classes:
 - Independent; the qualities are completely independent of each other, which implies that a change in one quality value has no effect on the other.
 - Related; the qualities can be related with a parameter *ValueImpact*, which represents the strength of the relation (Weak, Medium and Strong), and the parameter *ValueDirection*, which represents the direction of the relationship.
- **AggregateQuality:** Represents a combination between several qualities. For instance, the parameter *PricePerformance* ratio combines Price and Performance.

5.3.1.2 QoS Policy

The policy language, which enforces the service providers' policies as well as the consumers' expectations, is fully based on the ontology introduced above. Both consumer and provider policies are defined in an XML specified XSD schema as a de facto standard for describing the agreements in the form of XML documents.

Provider Policies

The main elements of this policy language for expressing provider policies are:

- `<WsPolicy>`; This element represents the root element. It has a required 'name' attribute which must be identified by the name of the provider and the required type attribute must equal 'provider' indicating that what follows is a provider policy.
- `<Services>`; This element may contain a sequence of `<Service>` elements specifying thus each service that this policy applies to.
- `<Ontologies>`; This element contains a sequence of `<Ontology>` elements each referring to an ontology.
- `<QoSPolicy>`; This element's role is to capture the provider's advertised `<QoS>` policy service or set of services. For each quality specified, the attribute 'promise' indicates the level of commitment of the provider to the advertised policy. This promise can have the following values: `bestEffort`, `guaranteed`, `notSpecified`, `noGuarantee`.
- `<qValue>`; This element gives the policy details about the quality element, which is specified in the element `<QoS>` with the attribute 'name'.

As stated earlier, beside a provider advertised policy, this policy language provides the customer with the possibility to specify a preference policy for a set of services and ontologies and a set of QoS policies as well. However, in the context of trust assessment with regard to the commitment of providers in a collaboration, we shall focus on the provider advertised policy for the implementation of the `AgreementInterceptor`,

the interactions automatically (this module shall be detailed in Subsection 5.3.3).

- When all policy documents have been received and processed, the resulting information shall be exported in a form of a comma separated file `scenarios.csv`.

The implementation of the module `AgreementInterceptor` shall be illustrated by means of a simplified example of the provider policy, shown in Listing 5.14. For a service 'Service1', line 15 indicates the value details for the quality parameter *UpdateInfo* indicated by the element `<QoS>`. In this example, the values promised by the provider on the element `<qValue>` are: min, max, and unit.

Listing 5.14: Provider policy example

```

1 <WsPolicy ... name='Provider1' type='provider'>
2 <Services>
3   <Service name='Service1'
4     interface='http://.../s1?wsdl'>
5   <Service name='Service2'
6     interface='...'>
7 </Services>
8 <Ontologies>
9   <Ontology name='QoSOnt'
10    uri='http://.../owl/qos.owl'>
11 </Ontologies>
12 <QoSPolicy ontology='QoSOnt' methods='.'
13   services='Service1'>
14   <QoS name='#UpdateInfo'
15     promise='best Effort'>
16     <qValue>
17       <typical>7</typical>
18       <min>5</min>
19       <max>10</max>
20       <unit>day</unit>
21     </qValue>
22   </QoS>
23 </QoSPolicy>
24 </WsPolicy>

```

The module `AgreementInterceptor`, which is realized as a set of XSL Transformations (XSLT) for transforming the policy XML documents, browses the source tree of the agreement document and concatenates the relevant quality information, as shown in Listing 5.15, in the following order:

```

$qualityParameter = [
  [provider;serviceName;countParam;QoSname;QoSpromise;
  param:content;param:content... ]
]

```

In the given order, each field refers to the following:

- `provider` indicates the name of the provider.
- `serviceName` indicates the name of the service being evaluated.
- `countParam` indicates the number of details about the advertised quality parameters.
- `QoSname` indicates the top name of the quality parameter.

- `QoSPromise` indicates the level of commitment of the provider to the advertised policy. It may vary between *bestEffort*, *guaranteed*, *notSpecified*, or *noGuarantee*.
- `param:content` indicates the details of the quality parameter concatenated with the content of each detail. Note that this last field might be concatenated successively according to `countParam`.

Once this information has been extracted for a given service, it shall be stored in the file `scenarios.csv`, which is generated automatically each time a change occurs in the agreements documents. This file shall be used by the module `interactionInterceptor` for evaluating the quality of the interaction following the data structure:

```
$header = [
    ["service", "qualityParameter", "status"]
]
```

where "qualityParameter" is mapped to the resulting quality parameter from the `AgreementInterceptor` module.

Listing 5.15: Exemplary XSLT for extracting and representing the quality parameters

```

1
2 <xsl:template match="WsPolicy[ @type=' provider ']">
3   <xsl:copy>
4     <!-- Template for extracting the quality parameters -->
5
6     <xsl:variable name="root" select="WsPolicy"/>
7     <xsl:variable name="provider" select="$root/@name"/>
8     <xsl:variable name="serviceName" select="
9     $root/Services/Service/@name"/>
10    <xsl:variable name="serviceIndex" select="
11    $root/QoSPolicy/@services"/>
12    <xsl:variable name="QoSname" select="
13    $root/QoSPolicy/QoS/@name"/>
14    <xsl:variable name="QoSpromise" select="
15    $root/QoSPolicy/QoS/@promise"/>
16    <xsl:variable name="countParam" select="
17    count($root/QoSPolicy/QoS/qValue)/>
18
19    <xsl:if test="$serviceName = $serviceIndex">
20      <xsl:variable name="qualityValues"
21        select="concat($provider,',' ,
22                      $serviceName,',' ,
23                      $countParam,',' ,
24                      $QoSname,',' ,
25                      $QoSpromise,',' )"/>
26
27      <xsl:for-each select="
28      $root/QoSPolicy/QoS/qValue">
29        <xsl:variable name="param" select="
30        $root/QoSPolicy/QoS/qValue">
31        <xsl:variable name="content" select="
32        string($root/QoSPolicy/QoS/qValue/$param)/>
33        <xsl:variable name="qualityValue" select="
34        concat($param,',' , $content)/>
35        <xsl:variable name="qualityParameter" select="
36        concat($qualityParameter,',' , $qualityValues)/>
37      </xsl:for-each>
38    </xsl:if>
39  </xsl:copy>
40 </xsl:template>
```

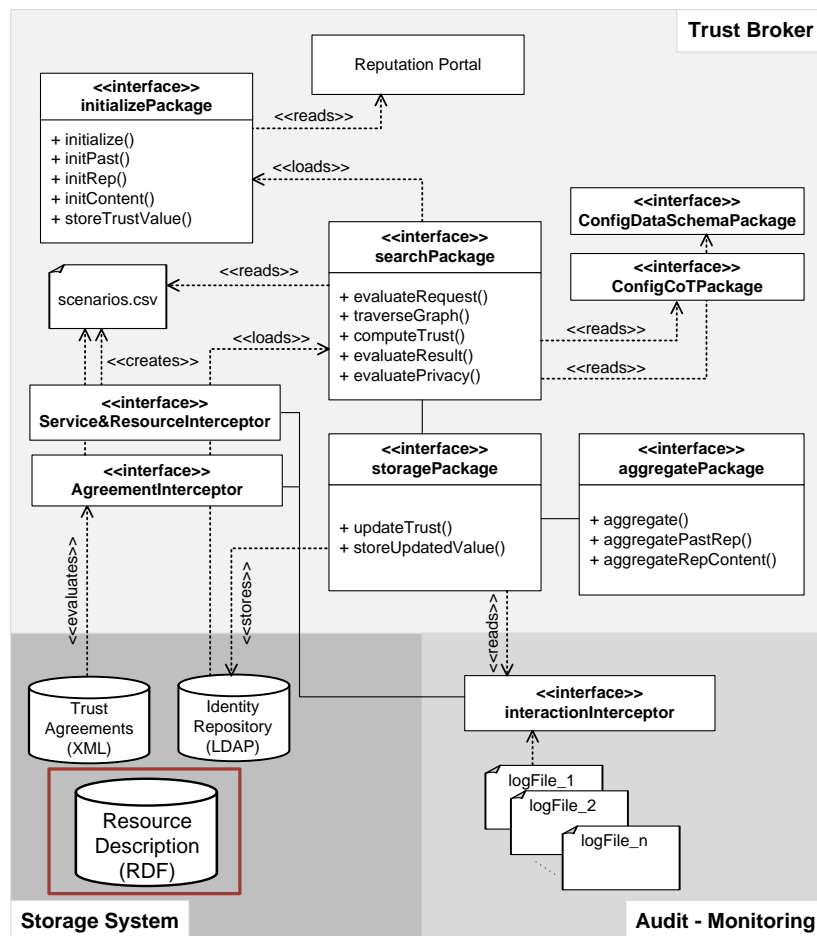


Figure 5.10: Resource Description Repository

5.3.2 Resource Description

Although the approach presented for service selection in Subsection 5.3.1 considers the trustworthiness of service instances based on user preferences and business policies with regard to QoS, this approach presents some limitations, because the ontology furnished therein is intended to be used in autonomic web services, and thus limited to the description of services.

However, the realization of the TBAC framework presupposes a richer knowledge representation for services, resources and qualities. Such representations help to capture the most important requirements to evaluate if the behavior of the principals is carried out as expected.

A complementary alternative solution for extending the representation of the different types of resources and services, which can be shared in the CoT and may be accessed by external users, is the usage of Resource Description Framework (RDF) [RDF] (an XML-based language for resource modeling). Similarly to the service ontology representation, presented in Subsection 5.3.1, the RDF information is required during the policy evaluation process. Figure 5.10 illustrates the extension of the Trust Agreement Repository with the Resource Description Repository.

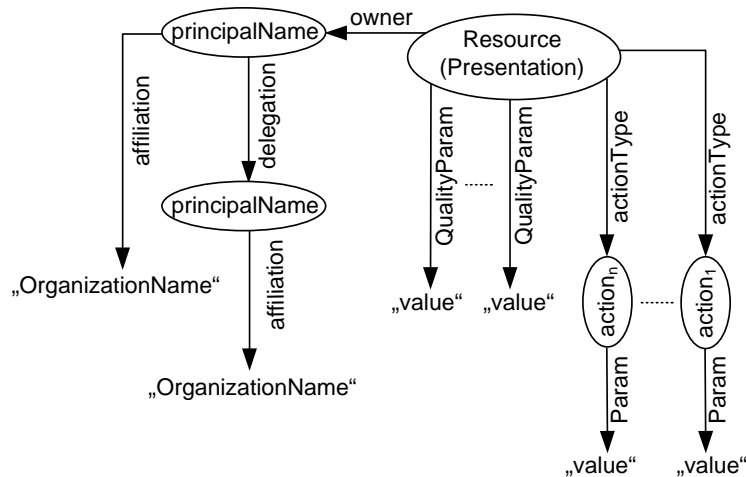


Figure 5.11: Resource definition in RDF

In this regard, resource descriptions can be related to several ITSM processes, such as configuration management, where quality insurance, trust and risk specific resource attributes may be added to their representations, e. g. as configuration items in an ITIL CMDB. However, no widely deployed standards exist for this purpose, so by using RDF extension our approach becomes more generic.

5.3.2.1 Concrete representation of the resources in RDF

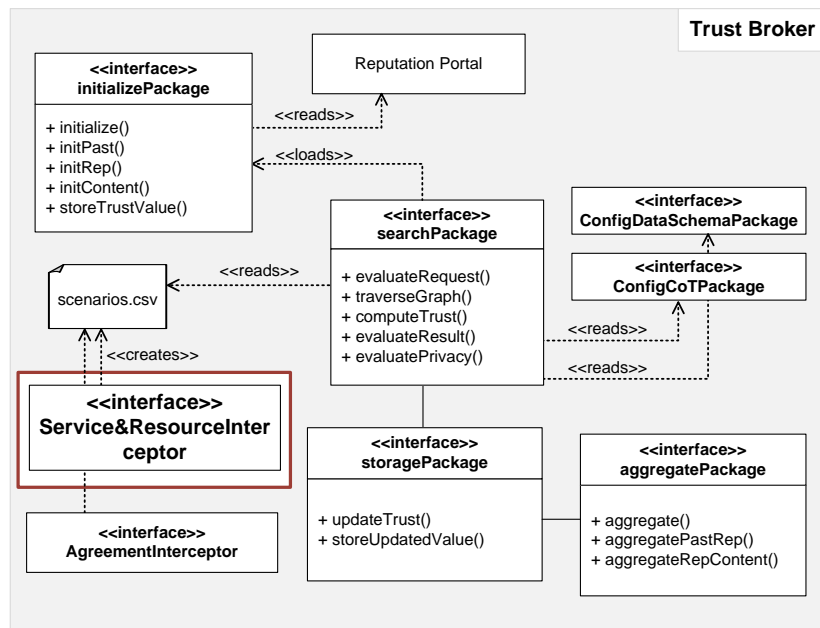
In RDF, resources are identified by URIs and have *properties*, similar to LDAP attributes. These properties associate the resource either with values or with other resources, which in turn have their own properties. Resources are identified as nodes and properties are defined as directed, labeled edges, which are also known as RDF arcs.

Figure 5.11 shows a generic representation of the RDF model for describing resources. In the context of trust management with regard to performance and quality, the properties related to the resource can be defined as *QualityParam*, indicating thus the quality value for a given quality parameter (the quality parameters can also be related to other quality parameters either in a sequential or hierarchical manner). Additionally, the resource can be associated with *actionType*, which represents each action that can be performed on the resource. The actions can be, in turn, associated with other properties, such as performance parameters or appropriate risk levels.

Using this approach, the complete content of the shared resources in the CoT could be described, including identity information, such as resource owner profiles (represented in *principalName*) which also can be associated with other resources or principals.

5.3.2.2 The module `Service&ResourceInterceptor`

The realization of the module `Service&ResourceInterceptor`, represented in Figure 5.12, is very analog to the module `AgreementInterceptor`. However, based on the RDF implementation, the effective use of metadata among several organizations within the CoT requires common data semantics, syntax, and structure. The

Figure 5.12: The module `Service&ResourceInterceptor`

Listing 5.16 illustrates a generic RDF description with appropriate name space specifications so that conflicts are efficiently prevented by defining a name space to avoid object name clashes between organizations and systems.

Listing 5.16: Exemplary resource definition in RDF

```

1
2 <?xml:namespace ns="http://www.w3.org/RDF/RDF/" prefix="RDF" ?>
3 <?xml:namespace ns="http://uri-of-name-space-1" prefix="CoT" ?>
4 <?xml:namespace ns="http://uri-of-name-space-n" prefix="NSn" ?>
5 ..
6 <RDF:RDF>
7   <RDF:Description RDF:HREF = "http://uri-of-Resource-1">
8     <CoT:qualityParam1 >... </CoT:qualityParam1 >
9     <CoT:qualityParam2 >... </CoT:qualityParam2 >
10    .
11    .
12    <CoT:owner >... </CoT:owner >
13    ..
14  </RDF:Description >
15  ..
16  <RDF:Description RDF:HREF = "http://uri-of-Resource-n">
17    <NSn:Property1 >... </NSn:Property1 >
18    <NSn:Property2 >... </NSn:Property2 >
19    ..
20  </RDF:Description >
21 </RDF:RDF>
  
```

In applying the RDF framework, built on XML, the initial ontology can be extended by defining an additional repository that recast domain-specific classes for actions and quality information and create appropriate instances. Accordingly, the implementation of the module `Service&ResourceInterceptor` extends the module `AgreementInterceptor` in such a way that additional XSL transformations enable the evaluation of the ontology used in the RDF documents. In the same manner, for instance, the quality parameter with its content can be read from the element

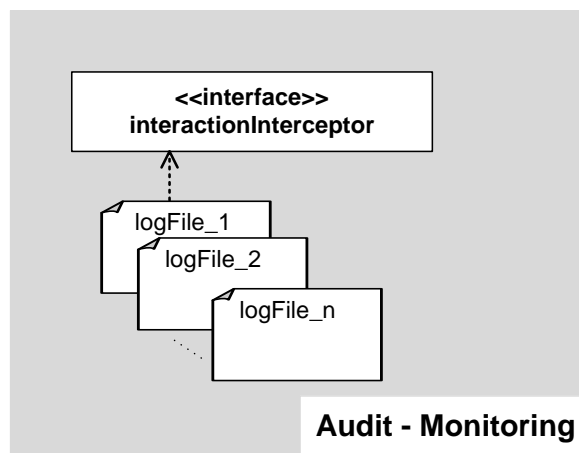


Figure 5.13: The module `interactionInterceptor`

<CoT:qualityParam1>, the resource name from the attribute `@RDF:HREF`, the resource owner from `<CoT:owner>`, etc.

5.3.3 Auditing the interactions

Obviously, having a bad experience with a communication or a collaboration partner can serve as an indicator of distrust for a future collaboration. That is, when there is a history of bad experience between entities, the trustworthiness, which can be interpreted from the quality of the interaction, tends to shape the weight of the trust relationship among the involved entities, and consequently influence their will for continuing or abolishing a collaboration. Further, when it can be proven that a collaboration rule or policy has been massively violated, the concerned entity can be, thus, regarded with suspicion and skepticism.

This is the basic idea behind the concept of the module `interaction-Interceptor`. As shown in Figure 5.13, this module is based on the information provided by the log files that are the records of interactions and activities among the principals in the CoT. Depending on the access and transfer protocols in use in the CoT, these records may require special tools to collect them. However, most application servers automatically record every transaction between the server and another computer in dedicated log files.

Log files usually include information such as the date and time of the transaction, a numeric identifier of the requesting entity, the resource that was requested or the actions that were performed on it, and most importantly the status of the request, for example the status that indicates whether the request has been successfully fulfilled, or if it ended with an error etc.

However, although log files may provide detailed information about how an interaction is closed, the statistics derived from them are not always easy to interpret, particularly if they are not combined with further informal search criteria. Based on that, in order to provide a more accurate assessment of how the interaction can be evaluated, the module `interactionInterceptor` shall be combined with both mod-

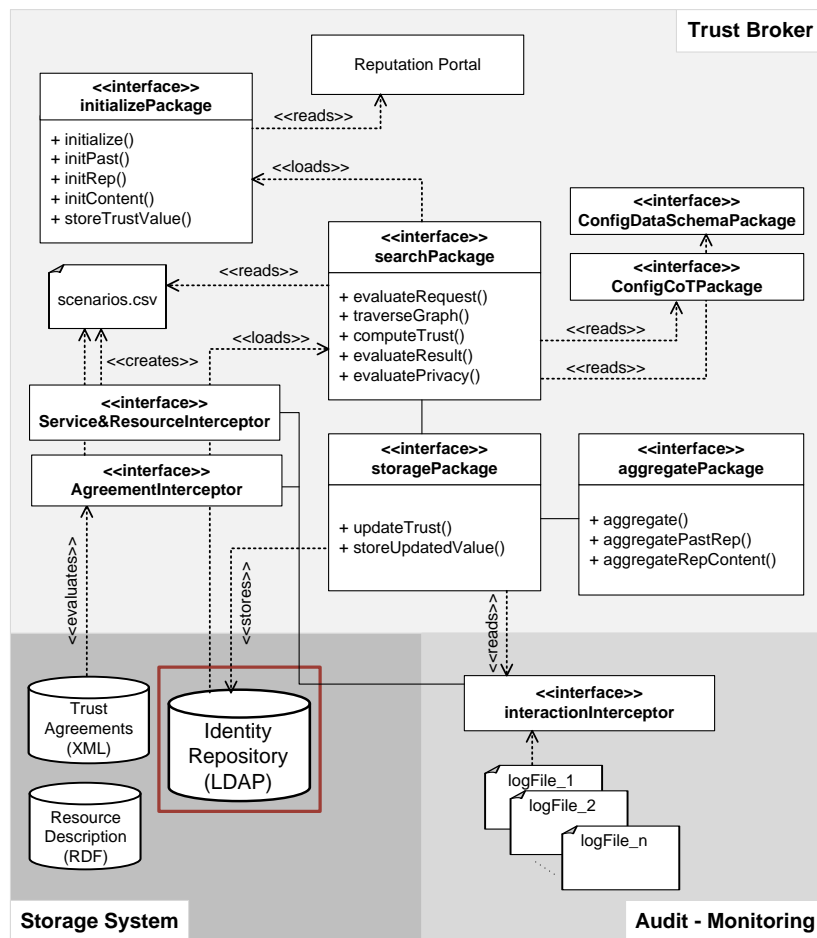


Figure 5.14: Identity Repository

ules `AgreementInterceptor` and `Service&ResourceInterceptor`.

By evaluating the quality and rule information, stored in the file `scenarios.csv`, the required search criteria for assessing the log files can be extracted from the field `qualityParameter`, where several quality parameters can be concatenated. Next, the status of the interaction shall be compared with the promised values in the agreements file. The result of this comparison may be either 1 or 0 depending on the degree of match between the two sources of information.

Finally, this module stores the results of the evaluation back in the file `scenarios.csv`, by extending the structure with a new field `status` (see below, both service and resource representations).

```
$header = [
  ["service", "qualityParameter", "status"]
]
```

```
$header = [
  ["resource", "action", "qualityParameter", "status"]
]
```

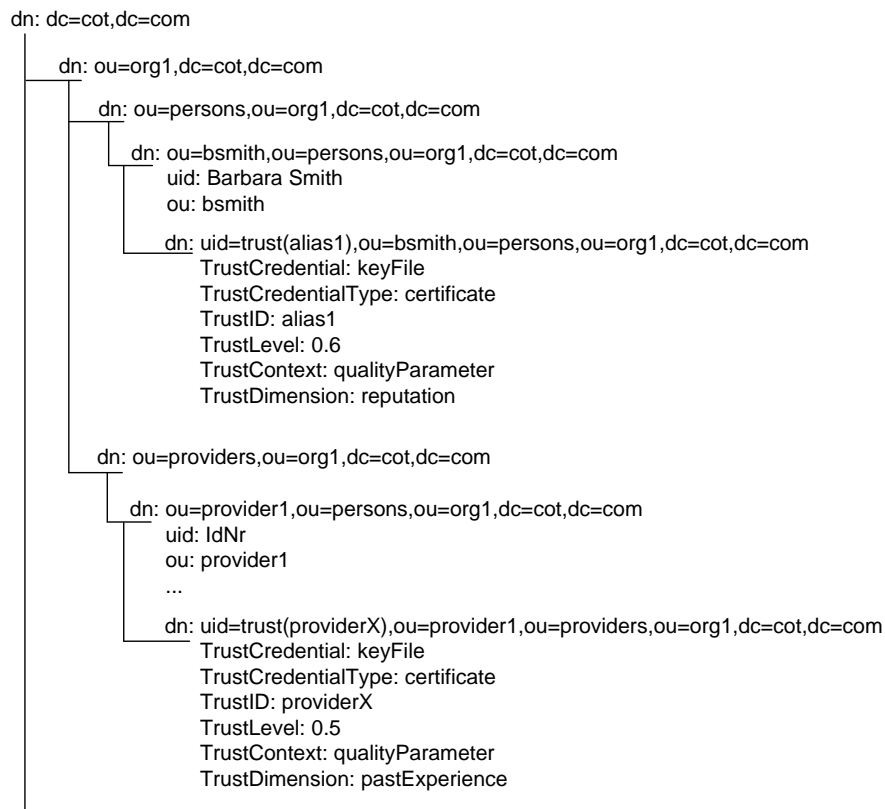


Figure 5.15: Principals data representation in a LDAP Directory

5.3.4 Identity Repository

The Identity Repository represents the last component in the storage building block (see Figure 5.14). Its implementation is built on top of previous work on TBAC [BD08], where the data model of the existing repositories to store the newly relevant trust has been extended. In this section, the LDAP schema extension implemented for identity repositories to store trust information shall be described.

Using LDAP for the prototype implementation in this chapter has shown many advantages, as the trust information is connected to entities, and most identity management solutions, which use this data, are based on LDAP directory services. This approach thus avoids the necessity of additional data repositories, which reduces the complexity of the overall TBAC architecture. Furthermore, LDAP is a standardized request-/response-based protocol, so the implementation is independent of vendor-specific drivers, such as those required for relational database management systems.

Data in LDAP servers is structured hierarchically and typically represented as a tree. The nodes of this tree are objects with an arbitrary set of attributes; each object is identified by its distinguished name (DN), which reflects the path in the tree from the object to the root.

As can be seen in Figure 5.15, principal objects include attributes such as the user's name, for trust relationships among persons. Depending on the collaboration scenario, the principals may also be providers, which are identified by other attributes (more

concrete examples together with evaluation aspects of this approach shall be detailed in Chapter 6).

For the management of trust among principals in the CoT, a new subobject `ou=trust{prefix}` is added; `ou` means *organizational unit* and is the standard structuring element for LDAP trees. To store the trust related data, a new LDAP objectClass `trustData` is designed. An arbitrary number of `trustData` objects can be assigned to each principal by placing them as leafs in the LDAP tree beneath the corresponding principal object.

Each `trustData` object has the following mandatory attributes, i.e. it cannot be created without specifying values for:

- `trustCredentialType`; specifies the types of credentials which have been submitted by the user.
- `trustCredential`; stores the submitted credentials. This is a structured data type (cp. [BD08]) which is stored BASE64-encoded in LDAP, similarly to other binary data types.
- `trustID`; indicates the principal being trusted. It can be deduced from the attribute `trustCredential`.
- `trustIDMember`; indicate the identifier of the organization to which the trusted principal belongs.
- `trustLevel`; indicates the level of trust for this principal on the specified subject area.
- `trustContext`; indicates the subject that the trust is about, either with regard to quality parameters if they are available or they specify the policy targets this object shall be applied to with regard to resources and actions on resources.
- `trustDimension`; indicates the method with which trust has been evaluated.

Additionally, in LDAP terms so-called optional, attributes can be used to store further details about the access and reputation history as well as recommendation chains if the user has been introduced by other known entities.

5.4 Access Decision Engine (ADE)

The resulting trust assessment as well as resource access rules will be integrated into an Access Decision Engine (ADE), which processes the information collected from the trust broker and triggers a policy decision point (PDP). On the one hand, the PDP decides solely based on the provided information, which also includes the relevant access control policies and environmental information such as the current date and time. On the other hand, it preserves the autonomy of collaborating organizations in maintaining their access control over the resources they share.

In summary, the role of the ADE engine can be divided into two main objectives:

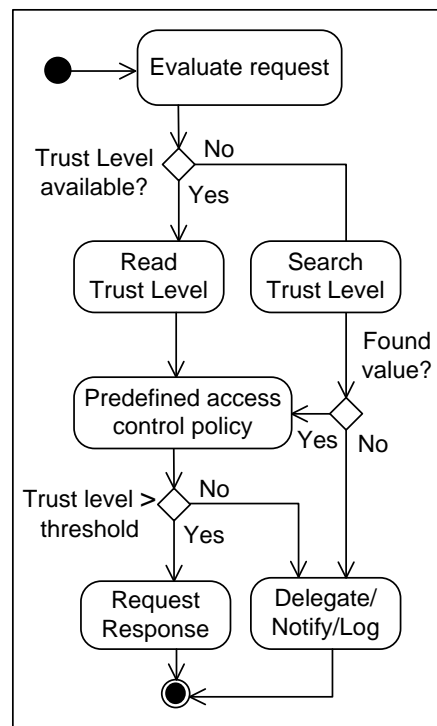


Figure 5.16: General flow chart representing access control decision with respect to trust information

1. Establishing *access decision* rules on the basis of the trust and risk information for the management of access decisions on the shared resources.
2. Protecting the *privacy* of the trust information; obviously, access to the trust information must be restricted, i.e. there must be a way to control which trust attribute a CoT member hands out to other members, in order to protect the privacy. The integration of privacy policies to support rules for the management of the release of such information is very essential of establishing trust in the CoT.

5.4.1 Access decision policies

Previously in several scenarios, it has been discussed that traditional authentication and access control are effective only in situations where the system knows in advance which users are going to access and what their access rights are. In the opposite case, the idea of using trust to provide finer-grained access control over the sensitive resources for helping to manage the security and privacy issues efficiently, has been introduced in this work.

This method helps a CoT member to determine whether not-directly known entities can be accepted based on their past behavior. In this regard, each CoT member can enforce access control policies, which can be expressed with trust levels as minimum thresholds (e.g., a requestor needs only a low trust level for accessing resource *A*, but a higher trust level for accessing a resource *B*).

The general flow of access control decision presented in Figure 5.16 shows that the

request query shall be transferred to the ADE engine. This latter initially checks if the requester is familiar in the CoT, by checking whether a dedicated trust level is available. If there is not any previous interaction correspondent to this query, now this module will perform a search method (based on the algorithms implemented in the trust broker). Finally, based on the trust value as well as on the predefined access policies and additional parameters (for example, risk management parameters, as discussed in Algorithm 4.4.2.3 in Subsection 4.3.3), a request response can be generated.

In this flow chart, access decisions are delegated to an external policy decision point, usually to the administrators in local domains, in two cases:

1. If the trust information made for the request is not available in the TBAC system.
2. If the trust information indicates that the principal is not trusted enough for the requested access right, particularly in cases where the risk level is too high.

In doing so, an additional layer of resource-local access control can be used to combine traditional access control mechanisms with TBAC, which is a typical prerequisite in real-world scenarios. The flow chart also demonstrates the use of two additional hooks. First, if the decision is *deny*, the user can be notified about the reason why his access attempt failed. Second, the access control result of all requests is logged to a tamper-proof database, which can, for example, be used for auditing purposes.

5.4.2 Privacy policies

The management of privacy policies for protecting and controlling the trust information flow among the members in the CoT has been widely discussed in Chapter 4 in Subsection 4.4.2.1. However, by mapping the trust information to resource objects, the same access control rules from *access decision policies* (in the previous subsection) can be applied here.

An applicable implementation of the PDP component, for both types of policies, can be handled by means of the eXtensible Markup Access Control Language (XACML) [Edi05], where the main elements `Subject`, `Resource`, `Action`, `Condition` and `Rule` can be effectively used for the purpose of this component.

5.5 Change Management

In Section 4.6, in which the last phase of the trust process model was analyzed, the arguments for the need of a change management process is brought forward. In the context of trust management, the change management often relates to modifications in the trust values, which reflect the change in the entities' behavior and performance. This change also relates to the agreements specifications, which often constitutes a consequence of change in other management disciplines.

- **Trust information:** Fluctuations of the number of entities (users and providers) entering and exiting the CoT, as well as their location in the CoT can have great impact on the correct functioning of the trust management processes. From the

Requirements	Fulfillment?
[Trust-Update]	✓ From the theoretical point of view, the fulfillment of this requirement has been discussed in Chapter 4, which is assured by the update function as well as the aggregation algorithm. In this chapter the implementation of the update function has been demonstrated.
[Rep-Update]	✓ The reputation values can be updated in the Reputation Portal. The influence on the final trust level is assured, in the same manner as for the trust values, by the update function as well as the aggregation algorithm.
[Sec-Update]	✓ The fulfillment of this requirement is assured by the extension of the trust agreements as discussed in Subsection 5.3.1.
[Risk-Update]	✓ The fulfillment of this requirements is assured in the RDF implementation, where the risk levels can be updated in the properties that are associated with the resources.
[Notify]	✓ The fulfillment of this requirement is assured in the ADE engine.

Table 5.14: Fulfillment of the remaining change management requirements

requirement analysis, provided in Chapter 2, the changes that regard the trust assessment process evolve around the trust levels, the reputation values, the risk information as well as the access decision policies. According to the results of this chapter, the fulfillment of these requirements is discussed in Table 5.14.

- **Trust agreements:** The agreements among the CoT members, e.g. SLAs, are subject to modifications as well. Some processes are quite susceptible to changes in operational agreements. For example, the introduction of new services or resources that shall be shared in the CoT may impact on existing service and resource repositories as well as on existing trust relationships. Obviously, the formal agreement specifications must be adapted to reflect that impact.

We argue that the usage of a unified QoS ontology, QoS policy language as well as a unified resource description (that follows standard name spaces in the CoT) for establishing the agreements' repositories (where the agreements are stored in a form of XML documents) can be easily adapted to the new items that can be added or deleted over time.

5.6 Summary and Conclusion

This chapter has addressed the functional components necessary to realize the implementation of the trust process model, presented in Chapter 4, within the TBAC Framework. A number of techniques were identified and described, along the life-cycle of trust relationships among the members in the CoT. The specific duties of each component as well as the interactions between them were subsequently described.

As discussed earlier, the TBAC Framework can be broken down into four main components:

The trust broker first is responsible for the creation and first-hand refinement of the information needed for the specification of the trust assessment process. This component encompasses complete process management workflows, starting with the initialization of the workflow part, taking into account implementation issues for representing and searching trust paths between principals located in distributed directories, and finishing with an aggregation module for the final representation of the trust information.

An important part of the trust broker has explored topics related to keeping trust information up-to-date and accurate (e. g., ways to recover from a bad reputation when freshly obtained trust information reflects a considerable increase in the confidence). A number of suggestions are made with respect to aggregation and update procedures for combining methods, which prove to be very useful for presenting a more complete assessment of how entities are interacting between each others.

The storage component, based on various forms of data schemes and structures, contains several storage sub-components and each with an individual purpose:

- Trust agreements repository is realized as a central component for drafting wills, engagement and other cooperation agreements among the members in the CoT. The agreement documents are realized with an intuitive QoS ontology and Policy language, which enable service providers to adapt clauses and content to fit preferences and partners' needs.
- Resource description repository complements the trust agreements repository, as it is intended to be used as a general method of modeling the resource information that cannot be described by means of the previous ontology. This extension requires, however, the producers of RDF terminology to agree on the semantics of resource identifiers by using common name spaces for standardizing the variety of syntax formats.
- Audit repository, which is based on log file analysis, serves as a data basis for reporting on the quality of interactions in the CoT. Taking the log files as sources of information into account, they can offer rich insights into the behavior of entities, because combined with other sources of information, such as the agreed upon rules as well as service and resource descriptions, they can be interpreted and used for assessing trust from past experience.
- Identity repository stores the trust information according to a dedicated scheme and a set of predefined privacy policies on this data. The design of this component shows a lot of flexibility, first, regarding the possibility of storing multiple objects of the same type without the need to extend the schema by defining new attributes

for each new created object. Second, it enables the search algorithm to browse the trees located in distinct domains effectively.

ADE Engine applies customizable access control policy to efficiently handle access rights for unknown principals, in such a way that only those principals with appropriate reputation and recommendation are allowed to gain sensitive resources.

Additionally, it addressed issues of combining the trust information with the risk information in trust-based access control. Based on the results of these components, several issues have been discussed such as the delegation of trust decisions and its automation, for example when it is an invalid assumption that a chain of intermediate entities exists which can be contacted on demand to acquire reputation information about the unknown entity.

In the next chapter, we evaluate the performance of our approach with respect to the accuracy of the obtained trust judgments, to the promptness at which audit information and reputation information is collected, as well as the adaptability of the model to the CoT member's distributed access control policies.

A proof of concept as well as exemplary use cases for the application of the TBAC Framework in real-world scenarios, which should be based on structural and real components, shall be proposed in the next chapter. These application examples help to concretize the purpose of the more abstract, functionally specified framework presented in this chapter.

Chapter 6

Evaluation and Performance Analysis

"Trust no one unless you have eaten much salt with him."

Cicero

Contents

6.1	Structure and notations	228
6.2	Comprehensive real-world scenario: Federated Learning Environment	229
6.2.1	Principals' roles	229
6.2.2	Overall interactions and relationships among the principals	231
6.2.3	Workflows between the three interaction types	233
6.2.4	Trust management issues and requirements	235
6.3	Applicability of the TBAC Framework	236
6.4	Performance analysis: What and how to evaluate?	248
6.4.1	Accuracy of the trust information	249
6.4.2	Trust Metric	253
6.4.3	Access Control	255
6.4.4	Integrability of the TBAC Framework	255
6.5	Discussions and Conclusions	257

The TBAC Framework investigated throughout this thesis may be appraised on the grounds that it is designed to be suited for use in different application scenarios, describing anticipated modes of operation for different classes of the CoT. In this regard, the primary goal of this chapter is to provide an evaluation of the prototypic implementation of the TBAC Framework, which as such must be assessed according to the benefits that can be reaped from its implementation. Conversely, the impact of its shortcomings must be estimated to allow discussion for future alternative improvements and extensions.

Following the sequence of the sections given in Figure 6.1, the chapter begins with a real-world scenario, which is chosen with care to reflect most of the CoT classes that

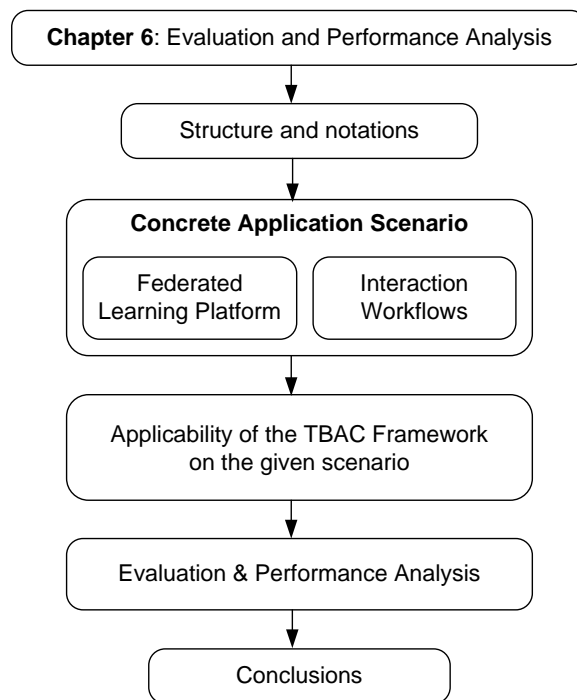


Figure 6.1: Sequence structure for Chapter 6

were discussed by means of three scenarios in the requirements analysis of Chapter 2. This application scenario helps to define in more concrete terms the problem that the TBAC Framework is addressing as well as its applicability to meet the requirements of the criteria catalogue.

Subsequently, an evaluation that aims at providing a proof of concept follows. By means of a performance analysis, it shows that the techniques, investigated in the previous chapters, can effectively ensure correctness of the trust information, which basically represents a serious problem in federated environments. Finally conclusions on the results of this evaluation close the chapter.

6.1 Structure and notations

From the definitions given in the previous chapters, it has been demonstrated that the CoT can be formed by connecting together dispersed individuals, groups, organizational units or entire organizations. In consequence, the possible structures are very much constrained by the available participants and the ways in which they can collaborate and can be connected together in a trustworthy manner. Usually these circles of trust can be understood as temporary or permanent coalitions of even geographically dispersed entities that can quickly come together to pool resources, capabilities and information to achieve common objectives.

Taking the concepts of the CoT into account, the negotiation, monitoring and enforcement of contracts and agreements with regard to trust are regarded as important components for the objectives of this work. Based on that, the scenario described in the

next section illuminates trust and contract management issues that will need to be resolved by the TBAC Framework to enable individuals and enterprises to operate with confidence.

To demonstrate the applicability of the TBAC Framework, we have chosen a comprehensive scenario that reflects the three known classes of the CoT (static, dynamic and virtual). Principally, this scenario is based on the concepts of the scenarios discussed in Section 2.2.

However, as a combination of all of these, it illustrates technical developments of web services and uses the concept of a Virtual Organization (VO) in grid technologies, which, for the purpose of dynamic collaborations, are rapidly removing the barriers around and between organizations.

6.2 Comprehensive real-world scenario: Federated Learning Environment

This scenario exemplifies a CoT of a federated learning environment. Applying the Virtual University of Bavaria and the IntegraTUM Project (see Section 2.2.1) as a basic scenario, some additional principles that enhance the provision of learning activities with regard to dynamic involvement of new participants as well as service selection and service aggregation are inspired from the European Learning Grid Infrastructure (ELeGI) project¹ as well as from the TrustCom Framework². In this context, the learner who is already registered with the Learning platform is assisted to define a customized training session according to his skills and personal preferences.

On the one hand, this scenario builds on the static aspects of the CoT because of the static number and duties of the participants in the collaborative learning environment. On the other hand, several dynamic aspects are emphasized through the dynamicity of the interactions. That is, a learner's request for a specific learning activity may initiate the formation of nested and dynamic services and resources, which belong to different domains and whose existence and evolution are bound to achieving the objective of the learning activity provision and enactment.

In this respect, for the objective of providing the learners with support during the whole cycle of their learning process, from the definition of objectives to the assessment of results through the construction of the resources (for example adequate course material) according to the advertised QoS parameters, several actors (principals) form the learning environment. As will be discussed below, this learning environment reflects the principles of a federated environment for a group of principals that are bound to the provision of a specific learning activity, and need to be assisted with a trust management system.

6.2.1 Principals' roles

In the following, the main principals, their roles in this environment as well as some key dependencies between them shall be enumerated. Two classes of the principals can be

¹<http://www.elegi.org/>

²<http://www.eu-trustcom.com/>

distinguished: The principals who are interacting with the CoT (persons) and those that are participating in the CoT, by providing or managing services and resources (CoT-Members).

Principals interacting with the CoT

- Learner is the entity that consumes the Learning Content Objects LCOs, which are provided by the training and service providers in the learning environments.
- Mentor is the entity that has some experience and can offer mentoring support to the other learners on some topics, usually for free.
- Freelance tutor is considered as a professional tutor offering his specialised mentoring support to learners about specific topics, but in contrast to the mentor, usually they require payment for the services they offer.

Principals contributing in the CoT (CoT-Members)

- Content Provider (CP) represents the provider that manages the LCO by constructing metadata schemes and mapping knowledge management ontologies to the LCOs.
- Identity Provider represents the repository that manages the storage and the update of the learners' identities as well as the identities of other parties.
- Knowledge Management Provider (KMP) represents the provider that is responsible for managing the concepts, the ontologies as well as other knowledge structures (for example domain-specific concept dictionaries) for representing and classifying the LCOs.
- Publishing Provider (PP) represents the provider that publishes the learning content stored in the content provider and that provides remote access to it.
- Tutor Agency Provider (TAP) represents the provider that manages the freelance tutor data (containing tutor profiles, skills, etc.) and provides searching facilities on this data.
- Training Portal Provider (TPP) represents the provider that provides the environment for learner to request learning material, learning experiences and tutor support activities.
- Discussion Forum Provider (DFP) provides virtual discussion forums between learners as well as between learners and their mentors or tutors. Additionally, this kind of discussion forums provide communication and collaboration tools.

In the following section overall interactions and relationships among the principals that are participating in the federated learning platform shall be outlined with the help of different models. Subsequently, Section 6.3, by means of activity diagrams, demonstrates in detail the applicability of the TBAC Framework for managing trust in these interactions.

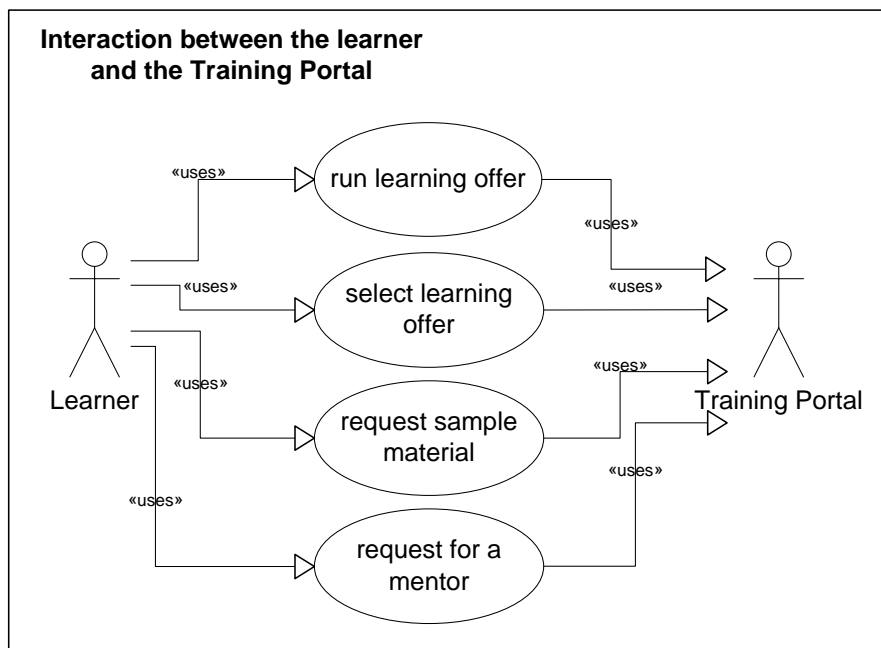


Figure 6.2: UML application diagram that shows an interaction: Learner–Training Portal Provider

6.2.2 Overall interactions and relationships among the principals

The possible interactions in this example can be broken down into three models:

- (1) Interactions between the learner and the Training Portal Provider.
- (2) Interactions between the participant providers that build the federated learning environment (CoT members).
- (3) Interactions between the learners and their mentors or tutors.

(1) Interaction: Learner–Training Portal Provider

Being registered in the learning platform, the learner shall be able to obtain a list of learning activities provided as services from the involved service and content providers in this environment. As discussed earlier, these services can be further customized or aggregated on the basis of personal requirements or preferences of the learner, because the learner has the possibility to negotiate the QoS of learning services, and can thus establish a formal agreement with the Training Portal Provider.

On the one hand, the learner interacts with the Training Portal, as illustrated in Figure 6.2, for example, (i) by requesting a set of activities specifying his requirements and learning objectives, (ii) by requesting for remedial work such as sample material of similar learning activities in order to improve his results, or (iii) by requesting help from other experienced mentors or tutors that have the required skills to provide this assistance.

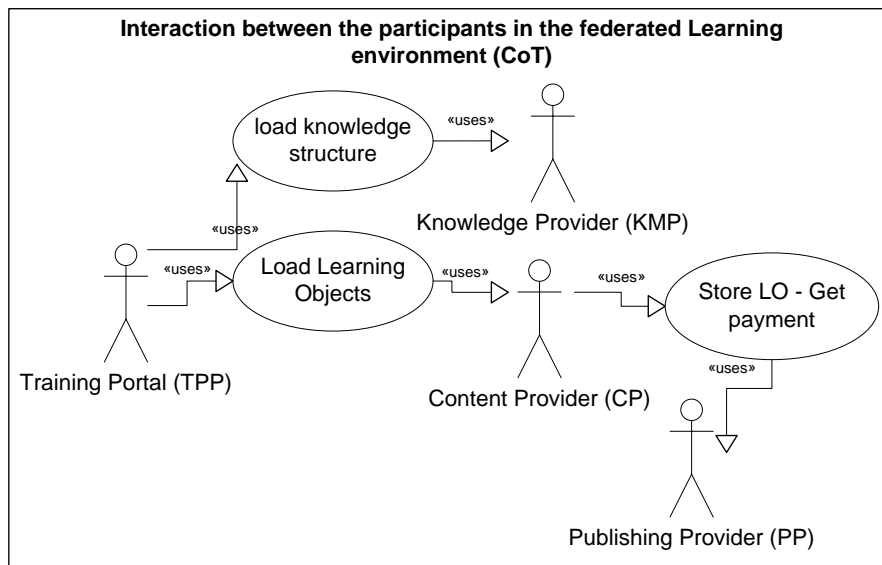


Figure 6.3: UML application diagram that shows an interaction: Provider–Provider (CoT members)

On the other hand, when the learner receives the requested learning activities such as the LCOs, he shall be informed about the constraints for using them such as the copyright issues and the payment conditions, if there are any.

(2) Interaction: Provider–Provider (CoT members)

The Training Portal Provider may be assisted by the Knowledge Agency Provider in order to analyze the learner’s requirements and preferences, and define a customized learning activity accordingly. In turn, the Knowledge Agency Provider, which represents the knowledge repository, is responsible for building knowledge structures like concept dictionary and ontology for mapping the learner’s requirements to the available learning resources.

An additional interaction example between two participating providers in the learning environment, shown in Figure 6.3, is that with the Content Provider, which builds the LCOs and manages the metadata that describe them, uses the knowledge acquired from the KMP. Moreover, the Content Provider can publish the LCOs into a separate Publishing Provider, which is responsible for the publishing activities as well as the management of payment for content provision of the Content Providers.

(3) Interaction: Learners–Mentors/Tutors

In most distributed learning scenarios, the learners meet each other in virtual meetings that are provided by discussion forums. In these discussion forums, the learners can discuss different topics on specific learning experiences with their mentors, who usually provide assistance without requiring payment, or with freelance tutors that usually provide professional assistance in exchange for payment as part of a reciprocal arrange-

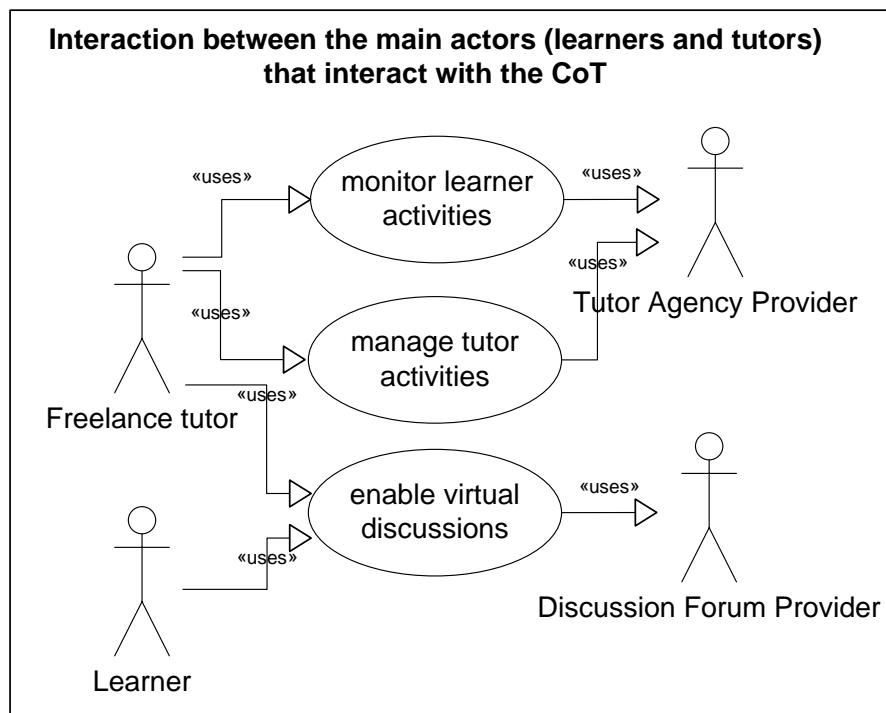


Figure 6.4: UML application diagram that shows an interaction: Learners–Mentors/Tutors

ment.

Further, the learners might want to send feedback about current or previous tutoring assistance. Figure 6.4 shows that the freelance tutor’s profiles and data activities are managed at a dedicated Tutor Agency Provider. The information stored at this provider relate to the collaboration tasks between the tutor and the learner, such as performance reports as well as other feedback information.

In the following subsection, concrete examples that illustrate the possible interaction workflows among the actors which might be involved in different interaction types shall be described.

6.2.3 Workflows between the three interaction types

We recall the scenario of the Virtual University of Bavaria (VHB) and exemplify the three interaction types among the known actors in the VHB as follows:

- The learner Bob, enrolled at the Technische Universitaet Muenchen (TUM) in the study program chemistry, may request a training session, at the VHB Training Portal, for visualizing an online demonstration course about experiences on the composition and structure of certain chemical reactions. For that, he performs a query to an automated broker at the VHB Training Portal, and consequently obtains a list of the offers that match his request.

The results shown on the VHB TPP side are obviously provided by different in-

stitutes such as the Chemical Engineering Department at TUM as well as other external multimedia content providers, which usually play the role of subcontracted third party providers for redirecting or conserving resources directed at the university departments for these particular tasks.

However, in his request, Bob might provide some QoS constraints. He wants, for example, to visualize only recent versions of the required demonstration course on multimedia servers that are known for their low prices and good visualization performance.

- Based on this request, the VHB Training Portal analyzes the requirements as well as the profile of the learner Bob and matches his QoS constraints and pricing options to the actual services that have been advertised by multimedia content providers to satisfy the given requirements. This match is based on the knowledge acquired from the provider KMP that is located at the VHB site, and on the identity information provided by the IDP that is located at TUM site.
- The result of the search will reserve the services that match the request. In doing so, a learning process path can be created by all intermediary providers. This learning process path starts with the VHB TPP, the KMP and IDP, the tutors that might provide assistance for the interpretation of the results of the experiences (they might also provide some support regarding additional disciplines that are related to the chemical experiences), and finishes with the selected multimedia CPs that need to be concatenated in order to meet the requirement of the requested demonstration course.
- As a result, the VHB TPP, on behalf of the involved providers, institutions and tutors that might be involved for enacting the learning activity, shall inform the learner Bob, and thus, negotiates an agreement with him for using these services.
- Once the agreement is established, the VHB TPP will confirm the availability of the learning activity that is represented in the multimedia file, so that Bob can start interacting with it. However, depending on the agreements (between the VHB TPP and Bob and between the VHB TPP and the content providers), Bob may directly interact with the content provider or may have access to the content via the VHB TPP, where the multimedia file has been transferred.

Obviously, all these workflows can be understood as a group of processes that follow a path leading to the achievement of the learning objectives agreed by the learner and the Training Portal Provider. They invoke and integrate dynamically several partners for providing the relevant LCOs. However, the agreement arranged between Bob and the VHB TPP does not necessarily include or specify the involvement of the other parties, such as an external multimedia content provider or a freelance tutor.

Since the relationships between Bob and these parties can only be regarded as transitive, in case the requirements of Bob have not been fulfilled as promised, or there is any violation in the collaboration rules, both Bob as well as the VHB TPP will be faced with severe trust management problems.

In the following subsection, the trust management problems and requirements for the given scenario shall be briefly sketched as an argumentation for the integration of our TBAC Framework in Section 6.3.

6.2.4 Trust management issues and requirements

Trust in this scenario is depending on a multitude of factors. Most importantly, contracting is the first factor that the collaborating parties in this environment rely on in order to formally establish and better manage their business relations. The agreements between the parties help to coordinate their business process activities by setting constraints and creating obligations to fulfill the collaboration objectives.

Based on that, the trust relationships can be established among the parties involved in a contract. Respectively, in the case discussed above, Bob has a long-term contract as a registered user, which allows him access to certain resources and services provided by the VHB learning platform. However, regarding the complexity and the interdependencies between the facilities of the learning activities due to the federated nature of the VHB, for each learning activity, a separate agreement might be negotiated between Bob and the VHB TPP as well as between the VHB TPP and the other parties, for dynamically setting up short-lived collaborations.

In this regard, the trust issues that often emerge in the given three types of interactions can be summarized in the following:

- (1) On the one hand Bob must be able to select the demonstration course provided by the server whose advertised QoS parameters better fit Bob's requirements. On the other hand, the VHB TPP, as the unique direct contact point for Bob, must be able to discover content provider needed to provide such a demonstration course, not only based on the match with the QoS constraints, but also based on their trustworthiness. For the purpose of achieving a simple representation of trust, this discovery functionalities must be enforced by the principle of trust levels in order to perform efficiently the search queries.
- (2) The other provider such as the KMP and the IDP as well as the tutors, which might be connected dynamically in this learning process path, need to have the possibility to choose and accept their collaboration partners as well. Moreover, they might also want to enforce fine-grained access policies, for example, giving access only to learners with special skills and good reputations. In the same manner as for the interaction between Bob and the VHB TPP, this selection process must be based on the trust level of the learner whose trustworthiness is subject of verification.
- (3) Once the demonstration course (realized in the multimedia file) has been delivered to Bob, Bob is additionally informed about the access to specialized groups of discussions, where Bob and other learners can discuss learning details with mentors and tutors. In these groups, the mentors might wish to add the interested learners to a group and to invite additional expert learners to a group. The major problem that the participants in these groups face is to securely establish collaborative activities for sharing information and experiences.

Additionally, since the groups can have a short lifetime (e.g. the time of a single learning session) or a long lifetime (e.g. a discussion built on multiple learning sessions), there is an eminent need for a trust management support for managing the relationships between the participants in the groups.

As a conclusion to these discussions, an appropriate trust, security and contract man-

agement solution is needed. This solution must include reputation and recommendation systems, mechanisms for the identification of service provenance as well as performance analyzers for assessing participant's compliance with the agreements that regard the collaborative tasks.

The interaction workflows, discussed so far in the VHB federated learning environment scenario, aimed at highlighting again the need for a trust management solution. As a combination of the three scenarios that have been investigated in Chapter 2 in Section 2.2, it exemplified, on the one hand, the static membership aspects for learner to register to the learning platform, and for service providers to join the CoT. On the other hand, it outlined the aspects of highly dynamic and ad-hoc collaborations among the members in order to achieve common goals and satisfy the learners' needs.

Based on that, we conclude that the same requirements that are summed up in the criteria catalogue (see Section 2.4.2) apply in this scenario. The next section will demonstrate how the TBAC Framework can meet these requirements and manages the trust relationships in the different interaction types.

6.3 Applicability of the TBAC Framework

In the previous chapter, it has been demonstrated that the principle of the TBAC Framework, as an application for all of these different interaction types, is to reason about trust in the following way: The initial trust relationships in the CoT is built up on the agreements, which specify the obligations and the duties of each partner in the federated environment. If one side fails to live up to one's part of the agreement, there is a *breach of contract*, which shall be automatically reported in the trust level of the concerned side.

However, the social network created between the principals interacting in the previous application scenario is obviously not restricted to one specific context and one communication model, but distributed among several contexts and models. Recognizing this distribution, in this section, a strategy of the TBAC Framework for interoperable relationship management, which is based on the different interaction models, is presented.

In this regard, building the concepts of the CoT on the existing federated learning platform requires first and foremost the establishment of an initialization phase, where the relevant information for the trust assessment process, the related data structures as well as the information needed for the configuration of the CoT are collected and prepared for the next phases.

As can be seen in Figure 6.5, the activity diagram indicates the action states within each component of the TBAC Framework that are represented as separate regions. The initialization phase visualized in this diagram, comprises the collection and setup of relevant information in the components Trust Broker, Storage System and the Access Decision Engine as follows:

1. In the Trust Broker, information that might be retrieved from the multiple dimensions of trust (trust by reputation from the reputation portal, trust from past experience and content trust from the audit system) are collected. In the example illustrated above, the trust information can be collected as follows:

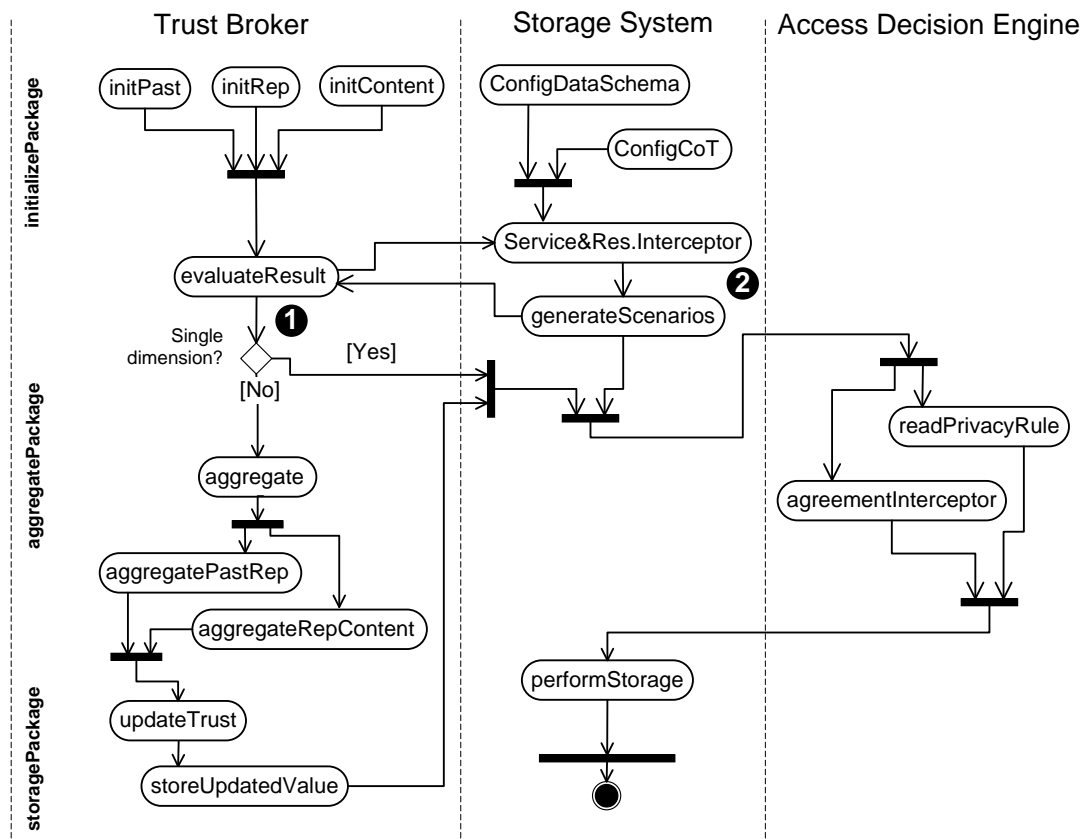


Figure 6.5: UML activity diagram that illustrates the initialization phase

- **Learner–Training Portal Provider;** Bob will have the chance to rate the multimedia content provider in the reputation portal of the VHB TPP, while the VHB TPP will have the chance to assign Bob with a trust level from the results of the audit system.
- **Provider–Provider;** all involved service and content providers on the learning process path shall be assigned automatically with a trust value, which results from the audit system.
- **Learners–Mentors/Tutors;** as stated earlier, all human actors in the VHB learning environment have the possibility to rate the performance of each other by means of the reputation portal.

The activity `evaluateResult` makes obvious that the trust information has to be correlated and aggregated in the `aggregationPackage`, when it generates from more than one dimension.

The decision point ❶ shows that in the absence of multiple dimensions, the collected trust value from one dimension can be stored according to the representation of the collaboration scenario that is specified by the module `Service&ResourceInterceptor` in ❷.

2. The second region of the diagram shows the initialization action states that

are the responsibility of the Storage System. It begins by setting the relevant parameters needed for the configuration of the VHB environment as well as the data structure for representing the shared resource and services (the multimedia files in our example), which, in turn, might be provided by different content providers, but must follow the same representation ontologies.

3. Based on the privacy rules as well as alternative constraints that might be defined by the module `AgreementInterceptor` in the Access Decision Engine, the storage of the collected trust information can be carried out in the dedicated repositories.

Once the CoT has been created for the VHB federated learning environment and initialized according to the illustrated activity diagram, it will have the ability to provide continuous support to the actors in this environment. As shall be discussed in the following sections, this support regards creating new trust relationships as well as searching and updating existing trust relationships with potential partners.

(1) Learner–Training Portal Provider

Due to the fact that the VHB TPP serves as an intermediary between Bob and the multimedia CP that can be dynamically integrated in the learning path process, the functionalities of the TBAC Framework for managing trust in the interactions between Bob and the VHB TPP can be applied to support both sides:

- (i) Bob for choosing only content providers that prove to be reliable with regard to the advertised QoS parameters that are represented as `recentVersion`, `lowPrice` and `visualizationPerformance`.
- (ii) The content and service providers for accepting applicants to access the content or to use the service only if they prove to fulfill certain requirements, for example the learner must have attended a related online course within the last two semesters.

As shown in the initialization activity diagram in Figure 6.5, the characteristics and the agreements on the available LCO (multimedia file) are first created and collected along with the foundation of the CoT (this initialization represents the beginning of phase 2 of the trust process model), and can be retrieved any time by means of the module `Service&ResourceInterceptor`.

Case (i): Trustworthiness of the Service Provider

The learner Bob specifies his QoS requirements on the VHB TPP site, which in turn, as discussed in Subsection 6.2.3, performs a search and sorts out the most suitable CPs that better match the QoS constraints. Based on the trust level of the candidate providers, a second sorting takes place, in such a way that only most trusted partners for the specified performance and quality parameters are going to be considered for the running collaboration.

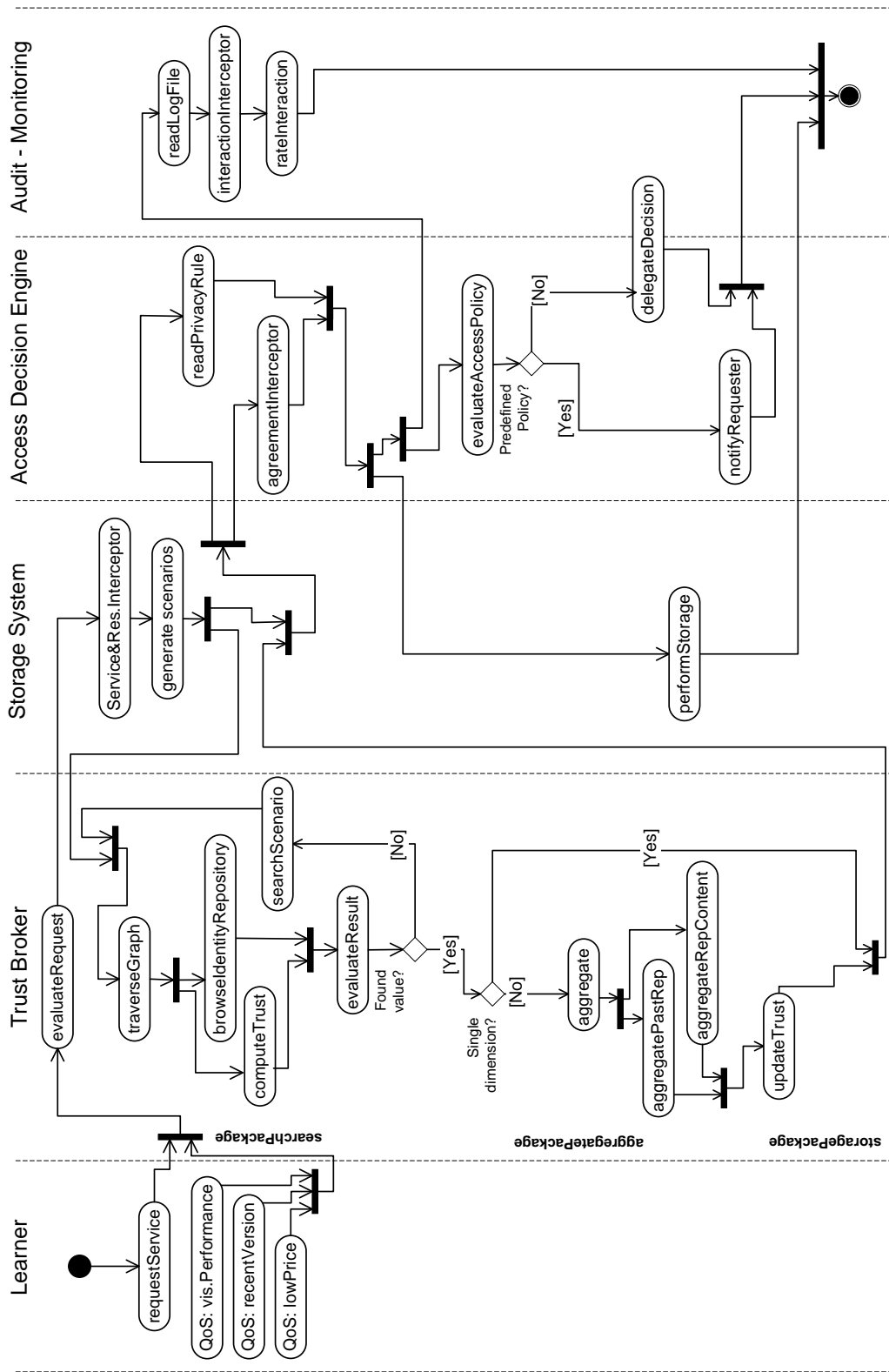


Figure 6.6: UML Activity diagram for the interaction: Learner-Training Portal Provider

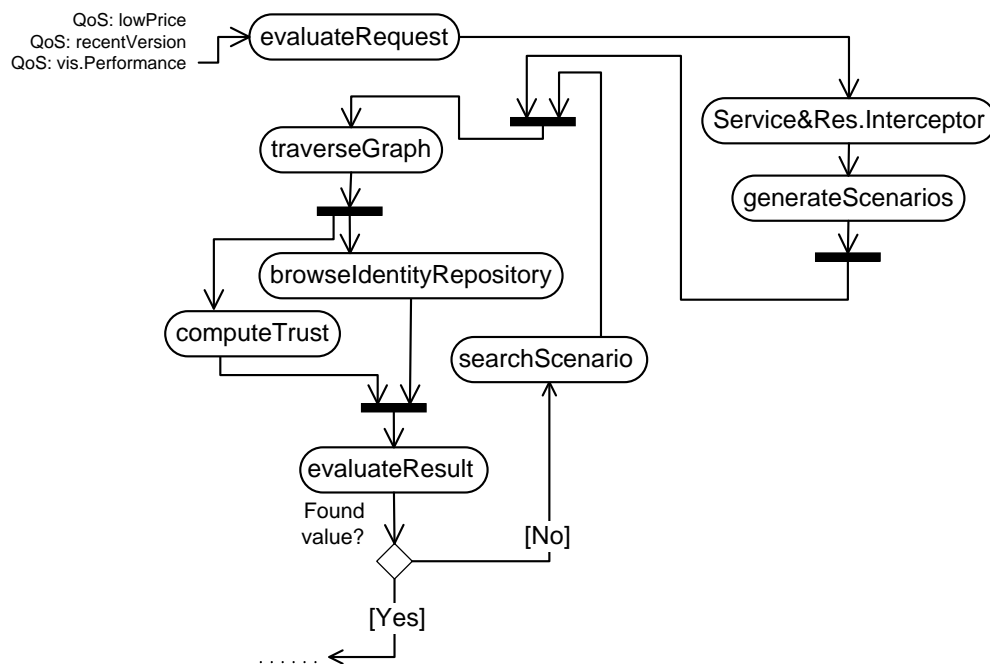


Figure 6.7: Segment from the UML Activity diagram for the interaction "Learner–Training Portal Provider" highlighting the `searchPackage` with regard to the QoS constraints provided by the learner

The activities that are invoked for this objective are represented in Figure 6.6. As can be seen in this activity diagram, for managing the trust relationship between the learner Bob and the VHB TPP with regard to the trustworthiness of the providers, the trust broker by means of the activity `evaluateRequest` evaluates the requirements of Bob and represents them as input parameters (`recentVersion`, `lowPrice` and `visualizationPerformance`) to the `searchPackage` of the Trust Broker.

This then - as shown in more detail in Figure 6.7 - later on maps these constraints by means of the module `Service&ResourceInterceptor` to the available resource and service descriptions, which are managed in the Service and resource repository (see Section 5.3.2) and thus finds a list of the content providers that advertised providing the requested multimedia file with the requested QoS parameters.

In addition to these results, the Trust Broker, on behalf of the VHB TTP, performs a second refined search in the Identity Repositories, which are located at each institution taking part in the VHB federation, and states the trust levels of the found content providers with regard to each advertised QoS parameter from past interactions. Obviously, it chooses the best match on each parameter for answering the learner's request.

Furthermore, the activity `evaluateResult` continues the search until all the scenarios that represent the QoS parameters have been investigated (see Figure 6.8). Suppose that the selected CP is a private Internet Multimedia provider for which the given task has been outsourced. For the given QoS parameters (`recentVersion`, `lowPrice` and `visualizationPerformance`), it has the following trust values (0.8, 0.6, 0.87).

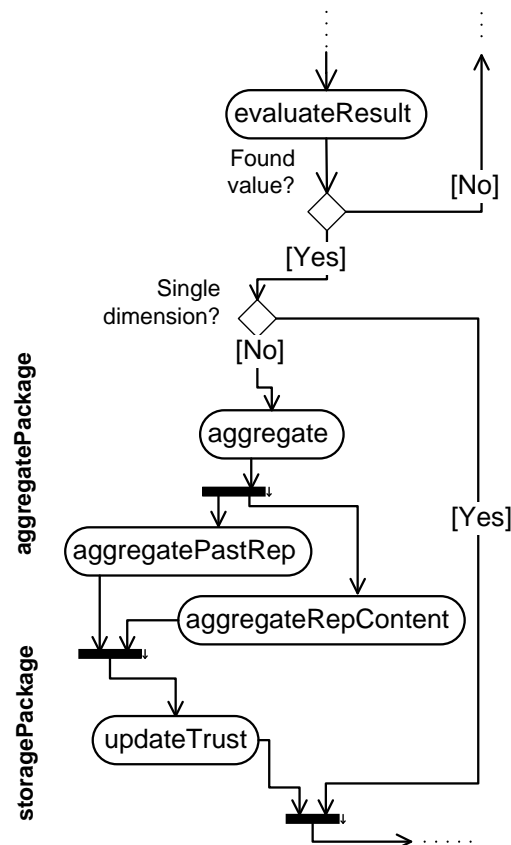


Figure 6.8: Segment from the UML Activity diagram for the interaction "Learner-Training Portal Provider" highlighting the aggregationPackage.

Next, this activity figures out whether the resulting trust values of this provider are not generated from other dimensions, for example from past experiences and - at the same time - by reputation. If so, it performs an aggregation according to the aggregation algorithm (see Subsection 4.2.4), and thus updates the trust values accordingly.

At the end of this workflow, the learner Bob will be notified about the results of the search and given the corresponding access information for the requested learning activity. Depending on the agreement between the VHB TPP and the other CPs, the required multimedia file and other related resources shall be provided at the VHB TPP site or Bob shall be given access to it directly at the CP's site. The interaction will be then audited for future requests.

Note that for this case the access control activity in the Access Decision Engine obviously does not play an important role, except for assigning access rights to the learner's personal data, because the basic idea behind these workflows is to support the VHB TPP for selecting the most appropriate content provider.

Case (ii): Trustworthiness of the learner

However, for this case the providers participating in the federated learning environment, following the same logic as in the learner's case, need to have the possibility to decide if

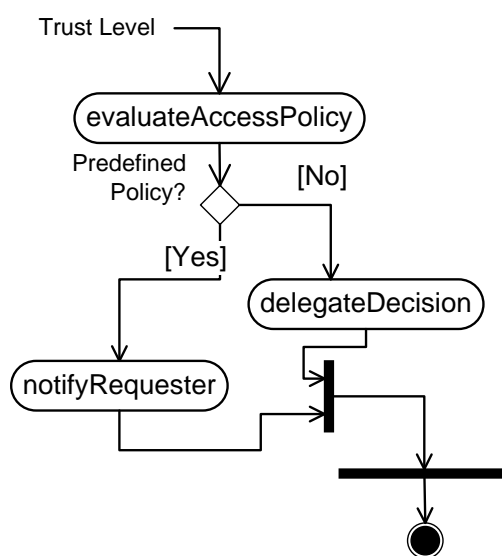


Figure 6.9: Segment from the UML Activity diagram for the interaction "Learner-Training Portal Provider" highlighting the Access Decision Engine.

the learner, according to the trust level that built on previous interactions, can be served or not. In this case, special attention is paid to the access decision activity as well as to the privacy protection and the agreements among the partners, which are evaluated at runtime by the module `AgreementInterceptor`.

In several federated learning platforms, experienced learners, after having made fairly long experiences with the learning environment, can be assigned with more rights such as the mentoring rights. In this regard, mentoring roles are regarded as structured (trusting) relationships that bring new learners together with more experienced learners who can offer guidance, support and encouragement aimed at developing the competence of the learner.

However, for Bob, as an experienced learner in the organic chemistry course, he wishes to gain mentoring rights for the online material for providing his assistance on related assignments in the discussion forum provider. Obviously, before the content provider could adjudicate him with these rights, Bob has to prove his competencies and trustworthiness in this matter.

The TBAC Framework in this exemplary relationship will assist the CP, the Chemical Engineering Department (that owns the course material), to find out whether Bob has the required degree of competencies in this subject, and furthermore if his trust level for the given requirements is beyond a certain threshold.

For example, the requirements of the CP on the learner in order to approve the mentoring rights can be mapped to the following performance parameters (scenarios in term of the Trust Broker): (*Knowledge, Commitment and Experience*). The trust level for all of these parameters must be greater than 0.75.

In the same manner as in case (i), Bob's request shall be evaluated and the trust level associated with the mentioned parameters shall be assessed in the `searchPackage`. Following the sequence illustrated in Figure 6.9, the resulting trust levels of Bob for the

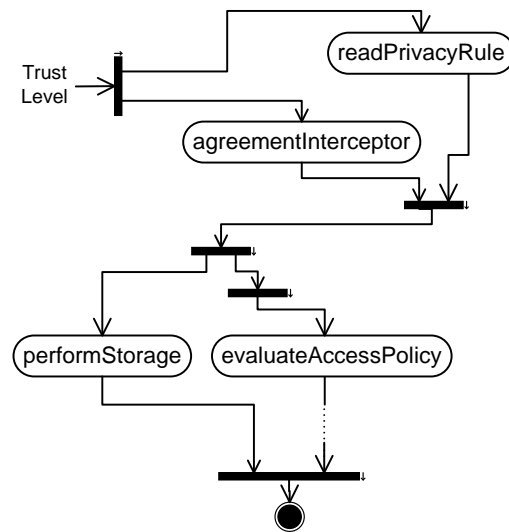


Figure 6.10: Segment from the UML Activity diagram for the interaction "Learner-Training Portal Provider" highlighting the `storagePackage`.

given requirements, are given as input parameters to the `evaluateAccessPolicy` activity in the Access Decision Engine. This activity shall compare these values with the CP's requirements, and decide about granting or denying access right on the course material with the mentoring rights.

Note that in case of uncertainties or incompleteness of the input parameters, this access decision might be delegated to the local administrators in the separate domains by means of the `delegateDecision` activity.

Figure 6.10 shows that for both interactions (i) and (ii), after a search has been performed, the resulting trust values are stored back in the corresponding repositories that are managed by the storage system.

(2) Interaction: Provider-Provider (CoT members)

The second type of interactions deals with the management of a trust relationship between two providers that might be assembled dynamically on the learning process path to fulfill certain learning activities.

Although the dynamic collaborations in this environment do not consider relationships between complete strangers, the aim of TBAC Framework, however, is to complement the relationships, which are created from the regulations and the agreements among the partners, by including ongoing trust assessment mechanism and standard-setting to takes care of the behavior of typically antagonistic parties for their mutual benefit.

In the VHB scenario, the visualization of the demonstration course involves a dynamic cooperation between the Chemical Engineering Department at the TUM site and an external multimedia Server, located at a private service provider. In this respect, the Chemical Engineering Department might have imposed some conditions before being automatically put on the learning process path. In the trust agreements it required, for

example, that the potential partner must regard the `Integrity` of the content with care as well as the `deliveryTime`.

The counterparty, the multimedia service provider, promises to cover the required QoS parameters. The representation of this agreement follows the presentation outlined in Subsection 5.3.1:

```
$header = [ ["service", "qualityParameter", "status"] ]
```

where the field `$qualityParameter` comprises relevant information about the quality parameter on which the agreement is based.

```
$qualityParameter = [  
  [provider;serviceName;countParam;  
   QoSname;QoSpromise;param:content;  
   param:content...]  
]
```

For this example, the `$qualityParameter` would comprise the following statements:

```
$qualityParameter = [  
  [multimediaServiceProvider;multimedia;  
   1;Integrity;bestEffort;IntegrityLevel:100%]  
  [multimediaServiceProvider;multimediaFile;3;  
   deliveryTime;bestEffort;minDelay:10s;  
   averageDelay:1min;maxDelay:60min;]  
]
```

Once the agreement is set, and the two providers have been put on the learning process path to fulfill Bob's request, the TBAC Framework performs a runtime evaluation of the quality of the interactions in order to ensure that the terms of the contracts and the commitments specified therein have been respected.

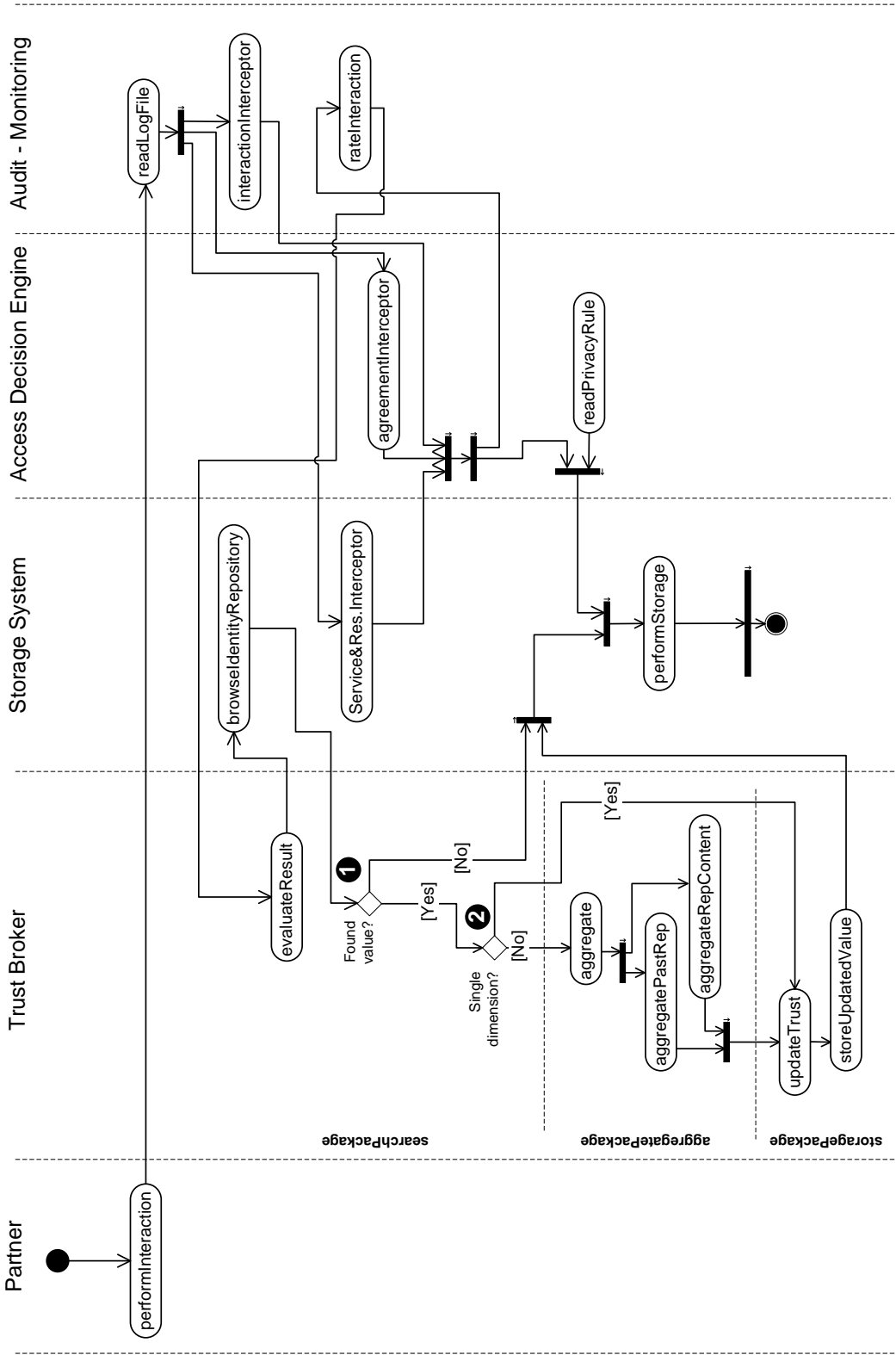


Figure 6.11: UML Activity diagram for the interaction: Provider-Provider

The principle is illustrated in Figure 6.11, where it can be seen that the interaction among the two partners is monitored by default (also any time the interaction within these constraints recurs). The monitoring in this case involves (i) the activity `readLogfile`, which collects the log files as a source of information, (ii) the activity `InteractionInterceptor`, which by means of the input information from the modules `Service&ResourceInterceptor` and `AgreementInterceptor` evaluates the interaction as reported in the log files and states the compliance of the multimedia service provider with the specified performance parameters (`Integrity` and `deliveryTime`).

Finally, the activity `rateInteraction` assigns a rating value to the interaction according to the result of the previous activity. For example, if the check on the MD5 sum reports that the integrity of the transferred file does not correspond to 100%, the interaction shall be rated with 0 for the QoS parameter `Integrity`.

In addition, as shown in the decision point ❶ in Figure 6.11, the trust broker, by means of the activity `evaluateResult`, will first check if a previous trust value for the multimedia service provider already exists. If so, it applies the update function for modifying the old value with the report of the audit system (trust from past experience).

Moreover, if the old value has been estimated by using another mechanism than trust from past experience (decision point ❷), an aggregation activity will then be carried out. This activity is implemented according to the aggregation algorithm presented in Subsection 4.2.4.

(3) Interaction: Learners–Mentors/Tutors

As with traditional face-to-face education systems, trust is seen as an important factor for the interactive distant learning systems, where strong trust relationships between tutors, mentors and learners can significantly improve the availability and quality of comprehensive and long-term learning programs.

Typical distant learning interactions among learners and their mentors and tutors or interactions among the tutors and the mentors themselves demonstrate that a trust control mechanism is necessary for motivating a growing number of participants to generate partnership and information sharing, especially in such non-profit federated environments.

For this type of interactions, which obviously requires the exchange of feedback ratings among the human actors, the TBAC Framework investigated in this thesis provides a reputation management mechanism that can be combined with the remaining components efficiently.

Similar to the previous activity diagrams, Figure 6.12 illustrates the activities that might be invoked when either a learner or the tutor leaves a feedback, rating thus the quality of the collaboration. This applies, for example, when Bob wishes to rate the compliance of the tutor to his support duties, or when, conversely, the tutor wishes to rate, for example, the punctuality of Bob in delivering the learning work at the pronounced dates, etc.

The principle here is very similar to the principle of trust from past experiences. The main difference can be found in the fact that the ratings are collected from the rating portal and are assigned by humans. Searching and retrieving existing trust values is expected as well, because the continuous evaluations from the audit system it may accumulate over time. However, the existence of previous trust values will accordingly demand an update as well as an aggregation activity.

So far the applicability of the TBAC Framework on the VHB federated learning platform has been demonstrated in detail by means of three interaction types among the actors that are either taking part in the information sharing or are just interacting by pulling information from the environment. In the following section, some quantitative analysis assertions of the effects of TBAC Framework in the performance of the existing environment shall be discussed.

6.4 Performance analysis: What and how to evaluate?

In this thesis, we have demonstrated through a number of scenarios that the trust information has become an asset as well as a vital tool for decision making processes. In federated and collaborative environments, in particular, the partners are often required to release some information they are already in possession of, in order to gain new information. Therefore, trust management solutions aim at supporting the involved partners to find out if the source with which information is exchanged can be trusted.

Due to the fact that the information being federated could be of a sensitive nature, the accuracy of the trust levels used for reasoning about prospective collaboration is of a great importance. This section addresses a set of criteria that is to be used to evaluate the trust assessment approach of the TBAC Framework, and to identify additional issues that can be addressed therein.

Concretely, this evaluation will concentrate on:

- Evaluating the accuracy of the trust information with regard to the update and the aggregation algorithms,
- evaluating the performance of the trust assessment process during establishment of trusted collaborations between parties as well as monitoring of the collaboration,
- evaluating the trust metrics used for representing the trust levels,
- evaluating the benefits that can be gained out of the Access Decision Engine, with regard to the flexibility of the automatic access decision, on the one hand, and the delegation of the access decisions on the other hand.
- evaluating the adjustment of the TBAC Framework on existing CoT technical interfaces, such as the alignment of the storage system with regard to representing,

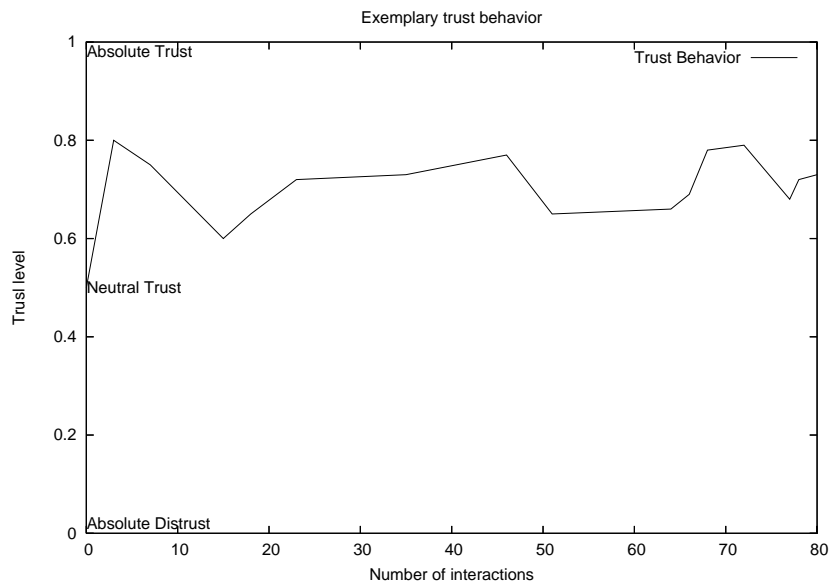


Figure 6.13: Exemplary trust behavior of a principal

retrieving and managing trust information in a generic manner.

6.4.1 Accuracy of the trust information

As discussed earlier, one of the most important features of the trust model that needs to be evaluated in this chapter is the accuracy of the computed trust level, especially when it is obtained from third parties while choosing the most trusted partners to collaborate with.

The TBAC Framework investigated in this work uses logical rules to analyze the nature of the interactions. A principal analyzes other principals it contacted earlier on and determines a trust level according to the result of the evaluation. As demonstrated in Chapter 4 and 5, the trust level is a single value that can help to control the interactions between participating principals, in such a way that values that are above a certain threshold are regarded as trusted and values below are regarded as distrusted.

The accuracy of these values is, however, very decisive. In order to test the algorithms used for computing and updating the trust levels, the following two aspects need to be taken into account:

- (1) The effect of malicious, old or erroneous recommendations on the trust level.
- (2) The accuracy of the trust assessment method in relationship with the structure of the graph as well as weight of trust relationships with intermediaries, i.e. the influence of the length of the path that relates two principals.

6.4.1.1 Case (1): Dealing with malicious ratings

It is obvious that if a principal assigns a negative rating value (trust by reputation) to another principal by mistake or by malicious intentions, the trust value of the concerned

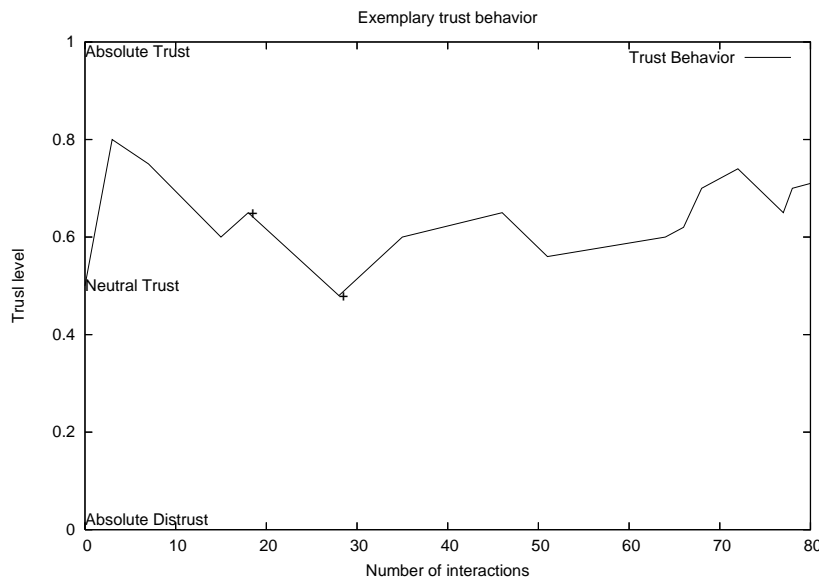


Figure 6.14: Influence of malicious ratings

principal may be diminished. In order to take this aspect into consideration, we perform the following test: We suppose the principal being rated in the CoT has already interacted with other principals, and thus the history of its behavior is recorded in the trust information repositories.

Figure 6.13 illustrates the exemplary behavior curve of the principal, where the trust level varies in relationship with the interactions.

Suppose that another principal tries maliciously to influence this behavior by assigning repeatedly negative ratings. We remind that the metric used for assigning the reputation values is based on the three levels:

- $\chi = 1$ for a positive rating
- $\chi = 0.5$ for a neutral rating
- $\chi = 0$ for a negative rating

Figure 6.14 shows how the trust level, in the trust curve, reaches from 0.65 down to 0.48 after 10 negative ratings according to the update function presented in Subsection 4.2.1.2. However, the difference observed in the trust value, shows that the update function, in contrast to many percentage computation algorithms, allows the trust level to decrease exponentially in function of the number of interactions.

This is due to the fact that the update algorithm does not consider the time factor, but instead the amount of interactions. It can be deduced that the longer a collaboration between the two principals lasts, the more the observed behavior is approaching the real behavior shown.

Another confidence aspect can be seen in the fact that all the interactions are monitored. That means, in case of malicious ratings, the aggregation algorithm always considers the audit information as the basis information for assessing the trust level, so that the

reputation value can be added or subtracted from the initial value according to the update equation.

However, the solution to this problem can be extended further by considering the following measures:

Convergence Factor

An important issue to consider the question when such malicious ratings are faced is the choice of the convergence factor α associated with the update function $(\frac{1}{2}e^{-\alpha(\sum interaction(x))})$ for coefficient updating. Due to the fact that the choice of the convergence factor affects the convergence speed of the trust level in relationship with the number of interaction, it has to be selected in dependence of the application scenario.

Confidence Factor

The definition of a confidence factor, which keeps track of the amount of positive as well as negative interactions used for increasing/decreasing the trust level can be helpful as well, because identifying repeated rating in the course of the interactions helps to approximate the correctness of the ratings. Note that this factor can be read off directly from the trust behavior curve.

In addition to these two parameters, the necessary of notifying the involved principals has to be considered, particularly when the resulting trust levels from the audit system and the reputation portal are contradictory, or differ to a large extent.

6.4.1.2 Case (2): Efficiency of the trust search and computation algorithms

While in case (1) the accuracy of the trust information with regard to malicious ratings has been discussed, case (2) concentrates on the efficiency of the search algorithms used for computing the trust level of a principal, whose trustworthiness is in question.

In Chapter 4, a set of search and computation algorithms from different dimensions of trust has proved to realize a dynamic trust model, which allows principals in the CoT to record a set of experiences that they will use later on in order to reason about a trust value. Experiences are updated and the updates influence further trust evaluation for principals in specific contexts. In addition to one's own experiences, principals have the possibility to gather recommendations from other principals and merge them in order to obtain a global and more specific analysis.

In this regard, it is worth mentioning that as more experiences are stored, the evaluation process becomes longer so that there is a danger of running out of space or time when dealing with large CoTs with large number of principals and interactions.

Therefore, the following subsections identify some evaluation criteria and pinpoint the issues that have to be addressed therein.

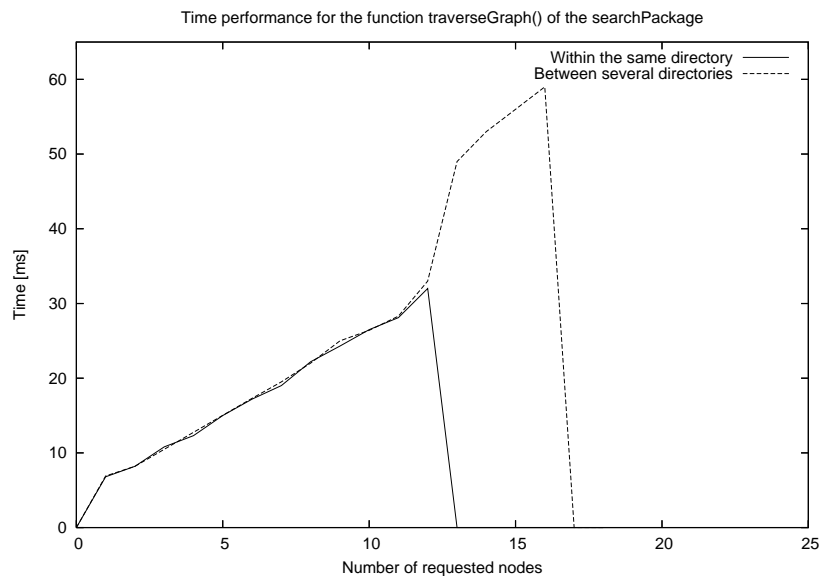


Figure 6.15: Time performance for the function `traverseGraph()` of the `searchPackage`

Time and Space Complexity

Subsection 4.2.2 has provided an analysis to the factors influencing the means by which the main search algorithm (Breadth-first search algorithm) is evaluated. We remind that this algorithm performs a search on the trust levels that might result from any trust dimension. By means of the tree structure of the graph and the search break condition, it could be shown that for a given CoT with n principals both the time and space complexity can be estimated with $O(n)$.

The principles of this algorithm are implemented in the Trust Broker within the `searchPackage` (see Subsection 5.2.2 for more details). To verify the validity and the correctness of this evaluation, we have conducted some measurement experiments by evaluating the execution time of the main function `traverseGraph()` including the calls it performs on the related functions that form the `searchPackage`.

The experiments that can be observed in Figures 6.15 are recorded into two separate sets in order to differentiate between those trust relationships that are a result of a direct edge between the requester and the requested node and those that are linked through intermediate nodes.

We simulate an acquaintance graph, in which the start node has 16 neighbors that are located in the same directory and two more neighbors that are located in other external directories. For performing the search, we simulate two cases: (i) The requester node is connected to one of the direct neighbors, and (ii) the requester node is connected to a node that is located in a different directory.

From the first graph, it can be seen that for the first case (where the nodes are located within the same directory), the curve starts to grow after about 8ms. This is due to the nature of the LDAP search, which first collects the results in a single array. Subsequently, the curve grows more or less linear until it reaches the search break conditions, as the edge to the requested node has been found at the principal object 15.

It is obvious that in the first graph the search is faster, as the LDAP queries needed in the second case can be spared, due to the fact that the desired trust value is stored within the same directory. Similarly, the second curve also needs an interval where a connection to the related LDAP servers is established. In the same manner the search breaks at node 17, indicating thus that the requested edge has been detected.

Both graphs reveal that the time runs linearly in relation with the amount of requested directories.

Search Break Condition

As stated earlier, the performance on the time and space complexity of the search algorithm is enforced by the feature of the search break condition.

The statements set in this condition (see Subsection 5.2.2.2) are confirmed by the previous experiments, because the search will break, either when a path between the original node to the target node is found, for which the time complexity clearly remains within $O(n)$, or when all the nodes that might be asked for a possible relationship to the target node have been visited. This represents the worst case of the algorithm and can be computed by exactly $O(n)$.

Further, the search break condition, on the one hand, controls the length of the search path that might connect two principals, and on the other hand, it compares the weight of the edges in such a way that recommendations from neighbors are taken into consideration only if their edges to the source nodes are stronger than to the requester node [Gol05].

For example, a principal A is looking for a trust value for requester C . A receives information from principal B that B 's trust value for C in context c is X . The trust value, the principal A has for principal B is Y in context c .

According to the `computeTrust()` function within the same package, A 's trust relationship to principal C in the same context as B 's relationship for the same principal can be adjusted with the value coming from B only if Y is greater than X , in order to more closely represent A 's own perceptions.

6.4.2 Trust Metric

Beyond the analysis we made with regard to the accuracy of the trust values' computation, the evaluation of the representation of these values is just as important. However, when looking at a trust representation, it is important to note that we are looking at the way trust is represented from a holistic point of view. In the following, we shall put forward arguments for our choice of the trust metric and discuss the advantages that can be derived therefrom.

In Chapter 3 varied models for designing trust metrics in different scenarios have been presented and analyzed. It has been discussed that the problem with all these wide and varied models is that there is no consistent set of criteria that is upheld throughout making it difficult for an interested party to decide upon a particular trust metric to implement.

Based on this analysis, we opted, in this thesis, for a combination of two discrete-

based-values metrics for the trust evaluation process, and an additional basic metric for the final representation of the trust levels. This combination is due to the fact that the trust values in our case studies may be generated from different aspects and dimensions, requiring thus different representation metrics.

While the basic metric for the final representation of the trust levels is designed within a continuous interval $T_l \in [0, 1]$, the metrics used for each single dimension are:

- **Trust from Past Experiences;** For this trust dimension, a very simple metric is used to evaluate an interaction, which may occur in various ways. Since the trust values from past experience are provided automatically by the audit system, either for assessing the behavior or the quality of the provided service or resource, only two status factors are needed to report about the way the interaction has been processed. Namely, 1 for a successful interaction, and 0 for a failed interaction.
- **Trust by Reputation;** This dimension also applies a simple metric, as it is intended to be used by end users (who exchange the ratings through the reputation portal). The corresponding three scales in this metric represent the main rating categories needed to report about the principal's performance and behavior, 1 for a positive rating, 0.5 for a neutral rating and 0 for a negative rating.

This representation proved to be very simple from the implementation perspective, and at the same time efficient for the final representation of the trust behavior of an entity, because the trust values, resulting from these dimensions, shall be scaled on the continuous metric by means of the update function.

In many trust models a low trust value does not distinguish between distrust as a result of bad experience and distrust as a result of a lack of information. Therefore, choosing the value -1 as an explicit parameter which reports about unknown status, solves this uncertainty in order to interact with a principal that has previously not been encountered, for example, no reports about past experiences with him have been obtained.

Context of the trust relationship

In addition to the trust metric, trust is entirely based on situations. A principal needs to take into consideration situational constraints before it chooses to engage in a collaboration. Such constraints may include different aspects such as reliability or quality of service.

Our trust model specifically takes the context into consideration when establishing trust relationships and dealing with other principals. In fact, context is seen as so important that the same principal will have varying trust values in different contexts.

In this respect, this trust model requires each principal to save the trust levels in value-context pairs (see schema representation in Subsection 5.3.4 for the storage of these pairs) so that when trust is determined it is not only the particular principal, but the context as well that is looked up.

One notable advantage that can be gained from combining the metric with context is the possibility of dynamic extension of these contexts. This is due to the fact that various forms of contexts can be chosen by the principal wishing to express any QoS or performance aspect within a context. Note that this expression is standardized through the use

of unique QoS ontology and policy language as well as a common RDF representation (Subsection 5.3.2).

Furthermore, this means of categorization allows for faster analysis than that of simple trust level representation because any member in the CoT can simply ask for category information (context) and make assumptions based on this.

Associating contexts with the trust levels obviously requires a higher processing overhead for the search process but is more effective when no static explicitly defined contexts exist, allowing members to create their own context as the collaborations in the CoT require. It is also possible to leave specific context for the trust level undefined and allow the trust value for a principal to be a context on its own.

6.4.3 Access Control

An important challenge with respect to the integration of the TBAC Framework within existing CoT interfaces is that principals, particularly, users are strongly concerned about their privacy. This is because in such open and federated environments, the partners might fear to loose control about who gains access to their federated services and resources.

Although this problem can be dealt with by appropriately defining access rules for privacy and relationship information, advanced concepts are however needed. In the Access Decision Engine (ADE) we provided an alternative solution to this problem, by classifying the access control decisions into two distinct categories: *automatic* with various restrictiveness and *local* access decisions.

For the first category, each member in the CoT can choose a set of actions or resources on which access decisions can be automated, and accordingly specifies its restrictions within the Trust Agreements Rules, in such a manner that if a certain number of conditions is fulfilled, the access may be granted.

Conversely, the members need to be able to designate alternative actions, whose execution might be critical. In this case, the ADE extends the metric of the trust information with the *risk* aspects when evaluating the result of an interaction with the predefined access decision rules. In case of uncertainties, e.g. if the risk level is too high, the ADE supports delegated access control administration, and forwards the access decision together with the collected trust information to the administrators in the local domains.

This solution, on the one hand, enables fast processing of requests through the automatic access decisions, and on the other hand, it preserves the autonomy of the members regarding performing critical actions on valuable resources.

6.4.4 Integrability of the TBAC Framework

In all cases of building or extending existing CoTs with the TBAC Framework developed in this thesis, there is a requirement for pre-existing cooperation contracts between the partners as well as common, standardized ontologies, process models and protocols.

In this section, the integrability of the TBAC Framework in an existing distributed trust management system for a given CoT shall be illustrated by using the example of the KeyNote system.

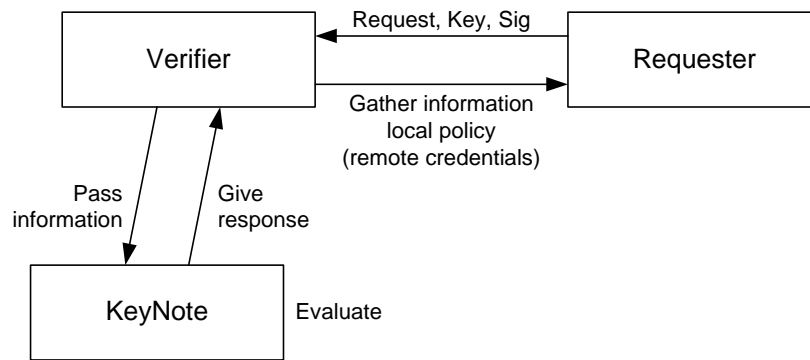


Figure 6.16: Interaction between an application and a trust-management system [IK03]

Extension of the KeyNote System

In Subsection 3.7.1 we introduced the access control model of PolicyMaker and its successor KeyNote [BFIA99] system, which in the context of trust management enable the expression of conditions and specify cases under which an individual or an authority can be trusted.

As illustrated in Figure 6.16, the KeyNote system, as an extension of PolicyMaker, accepts as input a collection of credential assertions and a collection of attributes that describe a proposed trusted action associated with a set of public-keys. Moreover, it requires that each principal which receives requests has a policy that serves as the ultimate source of authority in the local environment.

Typically the evaluation of these parameters and the decision whether compliance with the policy has been proven, is the responsibility of the KeyNote Engine. Subsequently, the result of the KeyNote evaluation process is a string, which indicates *authorized* or *unauthorized*.

However, since this system only extends the authentication of users' identities by specifying what a public key is authorized to do by evaluating whether a proposed action is consistent with a local policy. This authorization takes the delegation of rights from third parties into consideration as well. Following the trust definitions given in Subsection 2.1.2.1, we conclude that the KeyNote system covers only the dimension of *trust by delegation*. Consequently, the TBAC Solution presented in this work can efficiently complement this system with the remaining trust dimensions (see Figure 6.17).

Due to the fact that the KeyNote is realized as a simpler programming language (C-like programming language), its extension is thus very easy. We foresee this extension by integrating the functions of the Trust Broker as external libraries, which mainly deal with *trust from past experience* and *trust by reputation*.

Within this extension, the Trust Broker of course would require interactions with the other components of the TBAC-Framework, such as the storage components, however without interfering with the internal structure of the KeyNote System.

While the implementation of the TBAC-Framework has been studied and evaluated in detail, its extension with the KeyNote System is left for future work, because it requires large programming efforts.

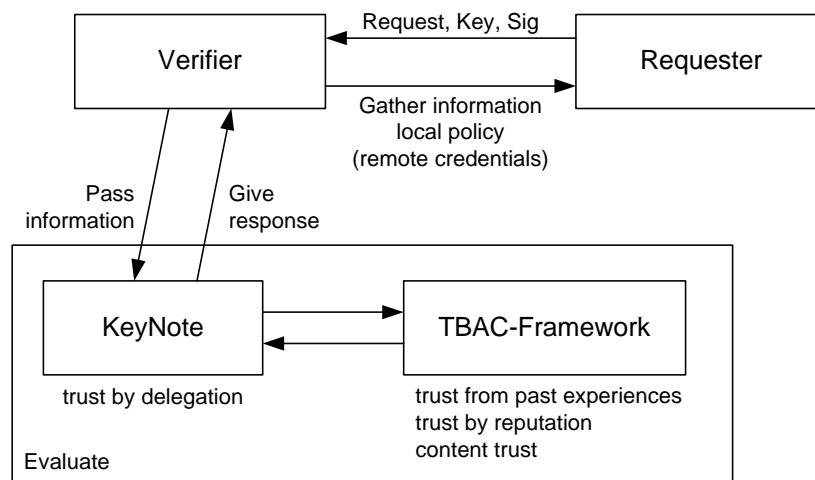


Figure 6.17: Extension of the KeyNote System with the TBAC Framework

6.5 Discussions and Conclusions

This chapter demonstrated, on the one hand, the applicability of the TBAC Framework on a real-world scenario, the VHB federated learning environment. On the other hand, it discussed a set of criteria that is to be used for the evaluation of this Framework and also to guide future improvements on the actual implementation.

These criteria have been intended to identify the worth of particular design and implementation decisions as well as the areas in which the TBAC Framework lacks attention. Using these criteria, we have been in the position to identify how our TBAC Framework addresses certain issues and also which issues have not been addressed. This knowledge is important when considering integration within other CoT interfaces and architectures such as in the Liberty ID-FF architecture or the KeyNote System.

Accordingly, the evaluation presented here showed from an empirical point of view that the TBAC Framework enables a certain level of trust to be established among collaborating service and resource providers in the CoT. It enables them to control each other's behavior by deploying unified trust, risk, resource and services as well as agreements metrics and representations, while respecting autonomy where each institution can retain its own policies on access control and conditions of use.

Obviously, the *proof of concept* of the TBAC Framework and the validity of the results presented in this chapter can be extended further through an evaluation of additional aspects that can be developed in this thesis. Such a thorough validation, however, would require more efforts in particular with respect to the realization of the reputation portal in order to conduct more experiences for evaluating the interactions among users with respect to performance, robustness and the fulfillment of user's preferences and requirements.

On the one hand, the evaluation of the assessment methods, used throughout this thesis enhanced a tradeoff between the accuracy of trust evaluation and a processing of experiments on the basis of use cases that have been required to perform this evaluation. On the other hand, we argue, however, that in order to achieve a more accurate trust evaluation, a more dynamic evaluation approach needs to be taken into account, which,

for example, continually incorporates changes in the environment and principal's interactions into trust evaluation.

Regarding the applicability of the TBAC Framework, the trust model considered for the VHB federated learning environment proved to be a very efficient basis for a number of possible variants. The generic aspect of this trust model certainly enables the implementation of additional learning scenarios to take strong benefit from the concept of the presented TBAC Framework. Conversely, this may demand additional features and capabilities, particularly, for learning environment that might be based on more enhanced collaborative, personalized and experiment-based learning paradigms. In this regard, the tests conducted through the prototypic implementation have yielded important hints for future work, which will be discussed in the next chapter.

Chapter 7

General Conclusions

"No matter how deep a study you make. What you really have to rely on is your own intuition and when it comes down to it, you really don't know what is going to happen until you do it."

Konosuke Matsushita

Contents

7.1 Summary of this thesis	260
7.2 Primary results and discussions	262
7.3 Evaluation of the criteria catalogue	264
7.4 Open issues and future work	264

The advent of the concept of federated environments has resulted in the exponential growth in interorganizational collaborations and, thus the need of availability of services and resources as well as information of the sources from which they can be gathered. This new perception of federated environments has caused a shift in the way static collaborations are conducted, as it strives to create a virtual presence and dynamic collaborations among the partners in order to attempt to take advantage of the inter-networked organizations.

Certainly, these new environments provide a wealth of new opportunities for gathering information, providing new services and participating in business interactions. However, in the same way that these environments provide new opportunities, they have triggered a similar interest in the concept of trust within the discipline of Computer Science, because the federation concepts often expose the participants to new levels of risk, and therefore increase the need for managing trust among them.

This chapter summarizes the contribution of this thesis with regard to the trust management problems in federated environments. In Section 7.1, it sums up the different parts of this thesis and discusses the main issues that have been sidetracked from the scope of the work investigated therein. Primary results from the contribution of the TBAC Framework as well as a number of follow-up ideas that can be cultivated, are presented in Section 7.2.

Section 7.3 concerns itself with the evaluation of the criteria catalogue. This evaluation regards the degree of fulfillment of the requirements studied in Section 2.4, which basically constitute the touchstone for the approach developed in this work.

Finally, Section 7.4 considers remaining issues that have not been solved in this dissertation and suggests some alternative directions for extensions and future work, especially with respect to the design of user interfaces, visualization, legal agreements and privacy management.

7.1 Summary of this thesis

The TBAC framework aimed at addressing the fundamental problem confronting federated environments throughout multiple sectors. It addressed particularly issues of making authorization decisions without possessing prior local knowledge of each requester in the environment. The number of situations where this problem will be unavoidable is enormous, and currently there is no satisfactory alternative solution.

However, a clear definition of trust and trust relationships has been hard to come by mainly because trust is subjective and can be regarded as a unique concept to each individual in each single scenario. Further, trust can be influenced by several factors, such as one's own beliefs, morals and experiences. **Chapter 2** introduced the concepts of the Circle-of-Trust that takes several trust definitions and dimensions into account, and suggested a formal model that can be applied for managing trust in federated environments.

Moreover, Chapter 2 selected three application scenarios that reflect the problem, which confronts any pair of principals attempting to establish trust with no prior contact or knowledge of one another, and showed the benefits of applying the formal concepts of the CoT on these scenarios.

For this objective, the scenarios have been selected with special care to exemplify three classes of the CoT:

- Scenario 1 illustrated a static form of the CoT through the static federation of the learning management systems (LMS), which is realized between two Munich Universities and the Leibniz Supercomputing Centre. This scenario highlighted the need for a trust management solution that manages the trustworthiness of online learners, which may require access to the learning content objects.
- Scenario 2 strived to complement the requirements on the static CoT environments with a dynamic form of the CoT. In order to do that, it illustrated the need of setup and management of dynamic trust relationships among the members as well as the users of the Multimedia Digital Library, which may consume as well as provide services and resources in a more autonomous and distributed manner.
- Scenario 3, building on the previous scenarios, exemplified the case where services and resources may be called upon in a task without previous knowledge of the other participants in the collaboration. Taking account of the concepts of the Virtual Organization in the DEISA Grid Project, the needs as well as the requirements for establishing trust between the involved partners in real time on a peer to peer basis have been analyzed.

At the end of this chapter, a summary of the requirements resulting from the three scenarios as well as from additional use cases for the management of the life cycle of trust relationships have been evaluated and weighted in a criteria catalogue. This criteria catalogue comprises 46 requirements that have been classified in different categories including direct trust, indirect trust, organizational, policy control and technical realization requirements.

Following the requirement categories of the criteria catalogue, in **Chapter 3** several existing approaches for the management of trust, reputation, access policies, and authorization have been reviewed in distributed and federated environments. The discussions on the applicability of these approaches to fulfill the requirements analyzed in Chapter 2 have led to an update of the criteria catalogue.

The criteria catalogue has served as a structured means in **Chapter 4** for the design and development of a trust process model as a solution to the problems investigated in the previous chapters. In distinction from existing approaches, this process model builds upon different phases, each of which has special tasks for the creation as well as the management of the life cycle of trust relationships, particularly, among principals that are not directly related to each other.

The five phases that shape the trust process model can be recapitulated in the following:

- *Initialization Phase* concerned itself with the definition of trust assessment techniques and workflows as the basis for any interoperability between the members in the CoT.
Within that, it focused on (i) the initialization of the trust relationships with regard to the notations and the required schemes such as trust-related attributes, metrics, query dimensions, etc, on (ii) the trust assessment algorithms that are based on the concept of trust from several dimensions, namely, trust from past experiences, content trust and trust by reputation, and on (iii) the aggregation algorithms that are needed for aggregation and representing the final trust values that may be assessed for a given context.
- *Storage and Management Phase* addressed the constraints for storing, distributing and accessing the resulted trust information among the members in the CoT. On the one hand, this phase proposed storage models, data structures and schemes for managing the trust information, and on the other hand, it suggested some alternatives for the realization of unified resource and agreement descriptions, which serve, consequently, as a basis for the following phase, particularly with regard to access rules and privacy issues.
- *Validation Phase* provided, in accordance with the previous phase, methodologies that can be followed for specifying rules with regard to the access and usage of the trust information. In addition, these access rules can be tuned to express the user's preferences and requirements in the service. In turn, they enable service providers, by means of the degree of trust that the participants assign to each other, to define their constraints for choosing their collaboration partners.
- *Evolution Phase* proposed tools for continual monitoring of the partners' behaviors in the CoT. By means of these tools, all continuously monitored changes in the behavior are to be echoed by changes in the trust of the involved parties according to the statement of the audited interaction.

- *Auditing and Change Phase* closes the trust process model, as it is aimed to complement the Evolution Phase by typically providing the continual monitoring with appropriate measures to be taken when changes occur in the CoT. These measures regard the changes in the trust information, such as the update and the aggregation algorithms, as well as changes in the agreements or the resource descriptions, which can be easily extended by using the ontology and the schemes already in place in the CoT.

Building upon the theory of the algorithms and the procedures presented in Chapter 4, **Chapter 5** has designed a Trust Based Access Control (TBAC) Framework for distributed relationship management in federated environments. This framework encompasses four main components:

1. The Trust Broker basically realized Phase 1 of the process model by implementing the algorithms as a set of Perl 5 libraries and packages that can be selected at runtime, because this component is intended to provide a means to encapsulate each algorithm as the function, and make them available when the demand arises.
2. The Storage System realized Phase 2 and part of Phase 3 by implementing and managing the storage of (i) the trust information in LDAP directories, (ii) the trust agreements in a QoS XML-Based policy language, (iii) the resource and service description in RDF, and (iv) finally the audit data in a dedicated XML scheme.
3. The Access Decision Engine realized Phase 3 within a policy engine. However, the integration of the XACML policy engine supporting Attribute Release Policies (ARP) for identity provider for controlling the release of users' data and Attribute Acceptance Policy (AAP) for service providers is a work in progress.
4. The Auditing Engine, in combination with the Storage System, realized both Phase 4 and 5 as a set of XSL Transformations that evaluate the audit files in contrast with the established rules in the trust agreements.

Finally, **Chapter 6** presented a prototypic implementation of the concepts developed in this work. Next to an evaluation analysis of some selected performance criteria, integration within the Liberty ID-FF architecture as well as the KeyNote System (RFC-2704) has been sketched as well.

7.2 Primary results and discussions

The approach presented in this thesis complements the static nature of Circle-of-Trust environments by a set of new dynamic trust assessment mechanisms. They support the exploitation of the existing trust base within the CoT while avoiding interference with single CoT members' local policies.

The main thrust of the approach, towards a more relaxed trust concept allowing for dynamically deducible trust values, can be emphasized in the following aspects:

- **Trust from several dimensions;** As stated earlier, trust can be seen as a knowledge gaining from several possible perspectives and dimensions. In order to cover

a larger view on the trust of a collaboration partner, the explored approach considered many dimensions that were categorized in collaboration trust as well as content trust, reflecting thus both personal experience as well as third parties' experiences.

The research findings on the deployed update and aggregation algorithms that merge the variable trust information revealed a high flexibility regarding the adaptability of the TBAC Framework to multiple application areas. At the same time, the study showed that this approach can be extended with new dimensions, if any.

- **Granularity - Trust Context;** All scenarios and case studies studied in this thesis have ascertained that the different collaborations in which a principal might be involved have to be separated into contexts. This feature of trust assessment undoubtedly enables to increase the quality of the offered services in the federated environment and to assist participants to express which aspect of others' behavior or quality they are interested in and at which level.
- **Adjustment of trust values;** For the regular, computed values, including an inertia regarding the change of trust levels, a means is provided, specified by an update function, to describe progressive effort for change in trust values. This function made high trust levels to be difficult to achieve, while adjustment of trust levels close to an initial trust value were made to be more dynamic.
- **Simplicity of representation;** In the context of assessing trust from several dimensions, this approach additionally enabled the use of both discrete and continuous metrics, both of which exhibit each dimension separately. By means of an update function, which has been explored to merge these metrics, the hypothesis of the update mechanism showed an additional aspect of flexibility and adaptability with regard to application scenario that might require one metric type, or even both.
- **Continuous Monitoring;** For the objective to gain an accurate assessment on the level of trust regarding a specific participant, continuous monitoring procedures have proven to be very useful for this aim, as they embody a continuous mode hypothesis and confirm or refute the hypothesis by unifying the established agreements with the observed behavior.
- **Assessment of the correctness;** Within the same context, regarding the accuracy of the trust information, additional measures have been proposed to enhance this aspect further. Defining confidence as well as convergence parameters, is made to extend the process model in order to serve situations, in which anomalous or malicious behavior can be detected.
- **Integrability;** The performed analysis on the previous aspects has shown the feasibility as well as the efficient integrability of the TBAC Framework within existing CoT platforms and interfaces, such as the integration within the standardized trust management system KeyNote for distributed relationship management.

7.3 Evaluation of the criteria catalogue

We recall the criteria catalogue and discuss the degree of fulfillment of the requirements according to the notation presented in the table below. This notation helps to delimit the contribution of our work, with regard to related works as follows:

- ✓:**Sota** indicates that the given requirement has been addressed in the so far discussed related works in Chapter 3 and is regarded as fulfilled.
 - ✓:**Chp4** indicates that the requirement is fulfilled through our approach. The proof of concept for the fulfillment of this requirements has been demonstrated in Chapter 4
 - ✓:**Chp5** indicates that the requirement is fulfilled through our approach. The proof of concept for the fulfillment of these requirements has been demonstrated in Chapter 5.
- (✓) a checkmark in parentheses indicates partial success in fulfilling the requirement with reference to the related chapter.
- (x) a cross indicates failure in fulfilling the requirement.

In both Chapters 4 and 5, the fulfillment in each single requirement category has been subsequently revised and compared with the outcome results of each chapter. Based on the discussions provided therein, we recall the criteria catalogue and sum up these discussions in order to provide an overall overview of the achievements of our approach in contrast with the requirements.

As indicated by the results of the criteria catalogue, the majority of the important requirements (weighted with degree 2) are satisfied by the approach presented in this work. The weights assigned to single requirements have been retained for reference in the table.

The requirements, which this approach - in its current form - does not provide the means to fulfill, are either left for future work, as is the case with the requirements [Risk-Metric] and [Content-Rep], or have been regarded as inflexible for integration purposes. This argument applies for the requirement [Tech- Protocol], which can be deemed to be not fulfilled, because the final design of the TBAC Framework made it independent from specific protocols or platforms.

7.4 Open issues and future work

As discussed earlier, our approach has sought a quick and cost effective way of setting up cooperations between CoT members and external organizations, without impacting third parties or compromising the CoT's integrity. Basically, it allows a trade-off between, on the one hand flexibility, speed and degree of automation in the setup of cooperation agreements, and on the other hand the level of security and privacy attained in such cooperations.

However, while the approach does fulfill the requirements, some related constraints must be taken into account when considering dynamic trust systems. In the following,

Direct Trust Requirements		Indirect Trust Requirements	
[SEC – AAA] (1) ..✓:Sota	[SEC – Policy] (1) ..✓:Sota	[Trust – Intern] (2)✓:Ch4, Ch5	[Trust – Level] (2)✓:Ch4, Ch5
		[Trust – Metric] (2)✓:Ch4, Ch5	[Trust – Context] (2)✓:Ch4, Ch5
		[Trust – Policy] (2)✓:Ch4, Ch5	
Trust by Delegation	Trust by past experience	Trust by reputation	
[Deleg – Auth] (1) ..✓:Sota	[Audit – Info] (1) ✓:Ch4	[Rep – Value] (1)... ✓:Sota	[Rep – Metric] (1)... ✓:Sota
[Deleg – TTP] (2) ..✓:Sota	[Audit – Metric] (1) ✓:Ch4	[Rep – Context] (1)... ✓:Ch4	[Rep – Cred] (1) .. (✓):Ch4
	[Audit – Eval] (1) ✓:Ch4	[Rep – Recent] (1) .. (✓):Sota	
	[Audit – Stor] (1) ✓:Ch5		
Trust Aggregation			
[Aggre – Collect] (2)✓:Ch4, Ch5	[Aggre–Scheme] (2)✓:Ch4, Ch5		
Interorganizational Access Control Requirements		Technical Realization Requirements	
[Access – Auth] (1) ✓:Sota	[Access – Policy] (2) ✓:Ch4	[Tech – Integrity] (2)(✓):Ch5	[Tech – Protocol] (2) x
[Access – Stor] (1) ✓:Ch5		[Tech – Storage] (2)✓:Ch5	
Organizational Requirements			
[ORG – TLA] (1) (✓):Ch5	[ORG – Cost] (0)✓:Ch5	[ORG – Integr] (1)✓:Ch5	[ORG – Impact] (1)✓:Ch5
[ORG – Time] (0) ✓:Ch5			
[ORG – Simple] (0) (✓):Ch5			
Policy Control Requirements			
Privacy Management		Risk Management	
[Priv – Collect] (1)(✓):Ch5	[Priv – Use] (1) (✓):Ch5	[Risk – Level] (1)(✓):Ch4	[Risk – Metric] (1) x
		[Risk – Rule] (1).....(✓):Ch4	
Change Management Requirements			
[Sec – Update] (2)✓:Sota	[Trust – Update] (2) ✓:Ch4, Ch5	[Risk – Update] (2).....(✓):Ch5	[Notify] (0) ✓:Ch4, Ch5
[Rep – Update] (2) ✓:Ch4, Ch5			
Content Quality Trust			
[Content – Quality] (1) ✓:Ch4	[Content – Rep] (0) x	[Store – Complex] (0)✓:Ch4	[Store – Monitor] (1)✓:Ch5
		[Store – Conflict] (1) ✓:Ch4, Ch5	

Figure 7.1: Fulfillment of the requirements in light of the criteria catalogue

we recognize some areas where the procedure presented in this thesis can be extended. Some of them ensue from fundamental problems, while others would increase the ease of application.

Extension of QoS Ontologies The Quality of Service (QoS) parameters, on which our approach based the concepts of *trust from past experiences* and *content quality trust* are merged along with a given QoS ontology and policy language to express the so-called contexts of trust. However, the expressiveness of this ontology is obviously limited, as it has been developed for specific service and resource usage scenarios.

Future research in this regard could be oriented towards the identification and the development of more advanced ontologies that deal with QoS and trustworthiness. The integration of new ontologies in the presented trust model and TBAC Framework should be a relatively easy task, since the modules that extract the trust information from these QoS representation have been designed for generic usage.

Interorganizational Trust and Service Level Agreements Another research issue arises from the fact that interorganizational trust agreements may conflict with the existing organizational security policies, SLAs and constraints. Future work concerning the TLAs should address the question of how to deal with these conflicts. In this regard, a formal study is needed to investigate what conflicting or inconsistent factors can be detected between interorganizational trust policies and local security policies.

Moreover, this future research should also investigate the way to systemize the verification process of the possible conflicts as well as the possibility of an automated detection mechanism that integrate all these aspects in an efficient manner.

Further trust dimensions Future research could be oriented towards the identification of further behavior trust elements. Their integration in the presented trust model should be a relatively easy task, since the model itself is very flexible and configurable. One additional dimension that could be integrated within the TBAC Framework presented in this thesis should consider the *belief theory* introduced in Subsection 2.1.2.1. Therein, the aspect of belief, as a further dimension of trust, corresponds to the case when all of the above defined dimensions of trust are missing, particularly when the entity is totally unknown to the CoT.

As an alternative, trust, in this vein, can be estimated by means of a theory and the expectations about the kind of motivations the unknown entity is endowed with regard to the shared services and resources in the CoT.

Business risk All trust relationships carry some form of risk factor with them. This applies not only to business context but any context where an exchange of information or service is required. In principle, the level of trust that is necessary in order to effectively cooperate depends, on the operational risk involved in trusting a foreign party.

Hence, the quantification of the trust level depends on the quantification of that business risk, which implies a dependence on the type, model and practices characterizing the business conducted by a given organization.

In this thesis, we introduced a simple means for handling this factor, by assimilating some form of risk levels into the decision-making process. However, more investigations on the means that could determine and handle the risk inherent in forming trust relationships is a vital component in a successful trust model.

More helpful ways of dealing with and controlling risk would require exploring fallback mechanisms to do some form of damage control that the worst case scenario could come about. In this respect, such fallback mechanisms would obviously require advanced knowledge of risk for making plans that take this risk factor into account and enables the partners to place constraints on it while requiring the interaction to take place within these predefined constraints.

Legal issues In several international projects for pervasive and federated service provisioning, it can be stated that the concept of federated environment in relationship with the Virtual Organisation has no formal legal meaning, as it is an informal term that builds on collaborations among peers.

However, when the virtual organization spans national borders there will be problems of jurisdiction, because individual partners may be constrained only by their local national laws, and this may further constrain the space of flexibility of the VO itself. We argue that integrating law as an instrument to achieve a Circle of Trust that is more secure and predictable, can be used to add strength to desirable rules of behavior in these environments.

Appendix A

Implementation Code of the searchPackage

```
1 package search;
2
3
4 use connect;
5 use config;
6 use Data::Table;
7 use File::Util;
8
9 my ($principalS, $principalT, $finalPrincipalT);
10
11 sub evaluateRequest{
12     ### Extract the requester information
13     my ($P1, $Org, $Px, $s) = readInfo();
14     return ($P1, $Org, $Px, $s);
15 }
16
17 searchScenario{
18     my($content) = $f->load_file('scenarios.csv');
19     @scenarios = split /(?!\\)/, $content;
20     return @scenarios;
21 }
22
23 sub searchPath{
24     ### Extract requester and requested IDs as well as the scenario
25     ### from the main function
26
27     my ($P1, $Org, $Px, $s) = evaluateRequest();
28
29     my (@trust, @principal);
30     my ($i, $j, $Tx);
31
32     ### Initialization ###
33     for($j=0; $j<scalar(@principal); $j++){
34         $principal[$j] = 0;
35         $trust[$j] = 0;
36     }
37
38     $Tx = traverseGraph($P1, $Px, $Org, $s);
39 }
40
41 sub evaluateResult{
42     my ($P1, $Org, $Px, $s, $Tx) = @_;
43     my ($T, $i);
44     my $header = ["T", "scenario"];
45     my $data = [ ];
46     my $trustValues = new Data::Table($data, $header, 0);
47
48     ### If the trust Level not found for the requested scenario
49     if($Tx eq "-1"){
50         ### Search the trust level for other alternative scenarios
51         @scenarios = searchScenario();
52         if(@scenarios){
```

```

53         foreach my $i(@scenarios){
54             $T = traverseGraph($P1, $Org, $Px, $i);
55             $trustValues->addRow([$T, $s],
56                 ($trustValues->nofRow)++);
57         }
58     }
59 }
60
61 ### Trust Level found for the requested scenario
62 if($Tx ne "-1"){
63     return $Tx;
64 }else{
65     return $trustValues;
66 }
67 }
68
69 sub computeTrust{
70     my ($principalT, $principalS) = @_;
71
72     my ($ID, $Tx);
73     my $N = 0;
74     my $M = 0;
75
76     my $tableSizeT = $principalT->nofRow;
77     my $tableSizeS = $principalS->nofRow;
78
79     for(my $i=0; $i<$tableSizeS; $i++){
80
81         my $P = $principalS->elm($i, "voucher");
82         my $ID = $principalS->elm($i, "ID");
83
84         for(my $j=0; $j<$tableSizeT; $j++){
85             if($principalT->elm($j, "voucher")==$ID){
86
87                 if($principalT->elm($j, "level") < $principalS->
88                     elm($i, "level")){
89                     $M += $principalT->elm($j, "level") *
90                         $principalS->elm($i, "level");
91                 }else{
92                     $M += $principalS->elm($i, "level") *
93                         $principalS->elm($i, "level");
94                 }
95             }
96         }
97         $N += $principalS->elm($i, "level") * $principalS->elm($i, "level");
98     }
99     $Tx = $M/$N;
100     return ($Tx, $P);
101 }
102
103 sub traverseGraph{
104     my ($P, $Px, $Org, $s) = @_;
105     my $i = 0;
106     my ($indicator, $PxLevel);
107
108     ### Get the information about the members from config.pm
109
110     ($Tx, $principalS, $principalT) = getEdge($P, $Px, $Org, $s);
111
112     $principalMiddleS = $principalS;
113     $principalMiddleT = $principalT;
114
115     if($Tx eq "-1"){
116         unless ($indicator ne $P1 || undef($principalS)){
117             ### Either a link is found or there are no more neighbors in
118             ### the lowest level of the tree
119
120             if(($principalT->nofRow) = 0){
121                 $principalMiddleS = $principalS;

```

```

123
124     unless ((( $finalprincipalT ->nofRow) != 0) ||
125             undef( $principalMiddleS )){
126
127         ### The same rule as described above
128         my $stableSize = $principalS ->nofRow;
129
130         unless ( $i >= $stableSize ){
131             ### Ask all the neighbors one level below
132
133             my $ID = $principalS ->elm( $i, "ID" );
134             my $MemberID = $principalS ->elm( $i, "CoTMemberID" );
135
136             ( $TxMiddle, $principalMiddleS, $principalMiddleT ) =
137                 getEdge( $ID, $Px, $MemberID, $s );
138
139             if( ( $principalMiddleT ->nofRow ) != 0 ){
140
141                 my $OrgPx = $principalMiddleT ->elm( 0, "CoTMemberID" );
142                 ### The index 0 is considered because $principalMiddleT
143                 ### can only contain one edge to Px
144
145                 my $PxLevel = $principalMiddleT ->elm( 0, "Level" );
146
147                 $finalPrincipalT ->addRow( [ $ID, $Px, $OrgPx, $PxLevel, $s ],
148                     ( $principalT ->nofRow ) ++ );
149             }
150
151             ### Collect principalMiddleS
152             $finalprincipalS = $principalMiddleS ->clone();
153
154             ### Make a copy of the table for the node $i
155             $row = $t ->delRow( 2 );
156             $principalMiddleS ->addRow( $row, 4 );
157
158             ### visit the next neighbor located at the same level
159
160             $i++;
161         }
162         ### Compute the trust level
163
164         if( ( $finalprincipalT ->nofRow ) != 0 ) {
165             ( $Tx, $indicator ) = compute( $finalPrincipalT, $principalS );
166         } else {
167
168             ### Consider the recursive call of the function traverseGraph()
169
170             for( my $j=0; $j < $principalMiddleS ->nofRow; $j++ ){
171                 my $id = $principalS ->elm( $j, "ID" );
172                 my $memberID = $principalS ->elm( $j, "CoTMemberID" );
173                 traverseGraph( $id, $Px, $memberID, $s );
174                 $PrincipalS ->addRow( [ $ID, $Px, $OrgPx, $PxLevel, $s ],
175                     ( $principalS ->nofRow ) ++ );
176             }
177         }
178     }
179 }
180 }
181 }
182 return $Tx;
183 }
184
185 ### Search in own container as well as in the container of other members.
186 ### $i helps to identify the position of the members whose parameters are
187 ### specified in connect.pm
188
189 sub getEdge{
190     my ( $P, $Px, $Org, $s ) = @_;
191
192     my $t = connect::create();

```

```

193 my %trustSchema = config::createSchema();
194 my $i = 0;
195
196 ### Search for the index of $Org ###
197
198 for(my $j=0; $j<($t->nofRow); $j++){
199     my $test= $t->elm($j,"name");
200     if($test eq $Org){
201         my $name = $t->elm($j,"name");
202         my $passwd = $t->elm($j,"passwd");
203         my $binddn = $t->elm($j,"binddn");
204         my $basedn = $t->elm($j,"basedn");
205         my $port = $t->elm($j,"port");
206     }
207 }
208
209 my $Tx = "-1";
210 my ($ID, $MemberID, $T);
211
212 my $header = ["voucher", "ID", "CoTMemberID", "Level", "Context"];
213 my $data = [ ];
214 my $principalS = new Data::Table($data, $header, 0);
215 my $principalT = new Data::Table($data, $header, 0);
216
217 ##### Connect to the corresponding CoT member
218
219 my $ldap = Net::LDAP->new($name, 'port' => $port)
220 or die "$0: The selected server: $name is not available!";
221
222 my $msg;
223 my $counter = 0;
224
225 if ($port!=636){
226     $msg = $ldap->start_tls();
227     $msg->code && die "$0: TLS with the LDAP-Server
228         '$name:$port' not possible!";
229 }
230
231 $msg = $ldap->bind($binddn, password => $passwd);
232 $msg->code && die "$0: Bind to LDAP-Server
233     '$name:$port' not successfull!";
234
235 my $key1 = $trustSchema{"type"};
236 my $key2 = $trustSchema{"context"};
237
238 my $search = $ldap->search(filter=>"
239     (&(objectclass=$key1)($key2 = $s))", base=>
240     $trustSchema{"namingAttr"."=".".P1." ,". $basedn);
241
242 foreach my $entry ($search->entries) {
243
244     $ID = $entry->get_value('$trustSchema{"ID"}');
245     $MemberID = $entry->get_value('$trustSchema{"MemberID"}');
246     $T = $entry->get_value('$trustSchema{"level"}');
247
248     if(($entry->get_value('$trustSchema{"ID"}') eq $Px)){
249         if($P == $P1){
250             print "Direct Edge to the requester is found!";
251             $Tx = $entry->get_value('$trustSchema{"level"}');
252         }else{
253             ### One of the neighbours has a link to the requester
254             $principalT->addRow([$P, $ID, $MemberID, $T, $s],
255                 $counter);
256         }
257     }
258     else{
259         ### None of the neighbors has a link to the requester
260         $principalS->addRow([$P, $ID, $MemberID, $T, $s], $counter);
261     }
262     $counter ++;

```

```
263     }  
264     return ($Tx, $principalS , $principalT);  
265 }  
266 1;
```


Appendix B

Implementation code of the initializePackage

```
1
2 package initialize;
3 use pastStructure;
4
5 use connect;
6 use config;
7 use Data::Table;
8
9 sub initPast{
10     ### Initialize the storage of the trust values from past experiences
11     my ($P1, $Org, $Px, $failedInteraction, $totalInteraction) = @_;
12
13     my ($ID, $Tx);
14     my $Failed = 0;
15
16     my $tableSizeF = $failedInteraction->nofRow;
17     my $tableSizeT = $totalInteraction->nofRow;
18
19     for(my $i=0; $i<$tableSizeT; $i++){
20         my $res1 = $totalInteraction->elm($i,"resource");
21         my $act1 = $totalInteraction->elm($i,"action");
22         my $param1 = $totalInteraction->elm($i,"parameter");
23
24         for(my $j=0; $j<$tableSizeF; $j++){
25             my $res2 = $failedInteraction->elm($j,"resource");
26             my $act2 = $failedInteraction->elm($j,"action");
27             my $param2 = $failedInteraction->elm($j,"parameter");
28             my $status = $failedInteraction->elm($j,"status");
29
30             if(($res1 == $res2)&&($act1==$act2)&&($param1==$param2)
31                 &&($status=="0"){
32                 $Failed ++;
33             }
34         }
35     }
36     $Tx = 1-$Failed/$tableSizeT;
37     return ($Tx, $P);
38 }
39
40 sub storeTrustValue{
41     ### Storage of the trust values from past experiences
42
43     my ($P, $Px, $Org, $Tx, $s) = @_;
44
45     my ($P, $Px, $Org, $s) = @_;
46
47     my $t = connect::create();
48     my %trustSchema = config::createSchema();
49     my $i = 0;
50
51     ### Search for the index of $Org ###
52
```

```

53     for(my $j=0; $j<($t->nofRow); $j++){
54         my $test= $t->elm($j,"name");
55         if($test eq $Org){
56             my $name    = $t->elm($j,"name");
57             my $passwd  = $t->elm($j,"passwd");
58             my $binddn  = $t->elm($j,"binddn");
59             my $basedn  = $t->elm($j,"basedn");
60             my $port    = $t->elm($j,"port");
61         }
62     }
63
64     my ($ID, $MemberID, $T);
65
66     my $header = ["voucher", "ID", "CoTMemberID", "Level", "Context"];
67     my $data = [ ];
68     my $principalS = new Data::Table($data, $header, 0);
69     my $principalT = new Data::Table($data, $header, 0);
70
71     ##### Connect to the corresponding CoT member
72
73     my $ldap = Net::LDAP->new($name, 'port' => $port)
74     or die "$0: The selected server: $name is not available!";
75
76     my $msg;
77     my $counter = 0;
78
79     if ($port!=636){
80         $msg = $ldap->start_tls();
81         $msg->code && die "$0: TLS with the LDAP-Server '$name:$port'
82             not possible!";
83     }
84
85     $msg = $ldap->bind($binddn, password => $passwd);
86     $msg->code && die "$0: Bind to LDAP-Server '$name:$port'
87         not successful!";
88
89
90     $result = $ldap->add( 'cn="trust($Px)",
91         $trustSchema{"namingAttr"}=$P1,$dn'
92         attr => [
93             'cn' => [ '$trustSchema{"namingAttr"}=$P1,$dn' ],
94             '$trustSchema{"MemberID"}' => '$Px',
95             '$trustSchema{"context"}' => '$s',
96             '$trustSchema{"level"}' => '$Tx',
97             'objectclass' => [ 'top', 'person',
98                 '$trustSchema{"type}"'
99             ],
100         ]
101     );
102     $result->code && warn "failed to add entry: ", $result->error ;
103 }
104 1;

```


List of Figures

1.1	Traditional centralized access-control architecture within a single organization	2
1.2	Relationships among research objectives	5
1.3	Process Model	10
2.1	Sequence structure for chapter 2	15
2.2	Federated Environments in conjunction with the CoT	16
2.3	Trust definition - Direct and indirect trust	20
2.4	Affiliation of the trust dimensions in connection with the position of the requester and the witness with regard to the CoT	21
2.5	Classification of trust dimensions for access control	22
2.6	Design of trust relationships	23
2.7	Basic relationships of trust definitions	24
2.8	Roles of the principals in relationship with the CoT	25
2.9	Agreements Rules in relationship with the CoT definitions	27
2.10	Resources being shared among the members in the CoT	28
2.11	Communication trust protocols and platforms of the CoT	29
2.12	Consortium model (Liberty Alliance Project)	32
2.13	Collaborative CoT (Liberty Alliance Project)	33
2.14	A dynamic federated environment for eLearning services	35
2.15	Extension of traditional centralized access-control architectures with trust management requirements for CoTs	41
2.16	Process model of the Digital Library case study	48
2.17	Presentation of an exemplary Digital Library open access model	52
2.18	An exemplary representation of the conceptual authorization hierarchy in DL	55
2.19	Requirements for indirect trust	60
2.20	Generalised application scenario for a Grid environment	66
2.21	Dependencies for Indirect Trust	75

2.22	Dependencies and weighting of the requirements leading to indirect trust	80
2.23	Criteria catalogue	85
3.1	Sequence structure for chapter 3	88
3.2	Indirect trust relationships with trusted third party (TTP)	90
3.3	Business example of the Liberty Circle of Trust	97
3.4	Liberty Trust Models	98
3.5	Sample deployment depicting actions such as the logging and the interaction with an auditor [CCD ⁺ 07]	102
3.6	Fulfillment of the requirements with regard to indirect trust in federated environments scenarios.	108
3.7	Fulfillment of the requirements in light of the criteria catalogue	123
4.1	Sequence structure for Chapter 4	126
4.2	Basic relationships of trust definitions	129
4.3	Representation of an exemplary trust behavior graph	130
4.4	Update function of the trust values	131
4.5	Criteria of the feedback ratings in eBay	132
4.6	interorganizational trust with respect to scenarios	133
4.7	Business and trust relationships	134
4.8	Representation of the trust acquaintance graph	135
4.9	Workflow 1 - Trust assessment within the CoT	136
4.10	Trust graph	138
4.11	Function <i>ComputeTrust</i>	140
4.12	A process traversing domains and VO	142
4.13	Agreements Rules in relationship with the CoT definitions	143
4.14	Fine-grained description of file storage actions	144
4.15	Representation of the trust values resulting from the audit system in the trust matrix	148
4.16	The breadth-first tree one gets when running the <i>ComputeTrust</i> on the given map by starting with P_1 to reach P_x	152
4.17	Feedback ratings in the eBay reputation system	154
4.18	Workflow2: Requester is not a member in the CoT	158
4.19	Phase 2: Represent, update and store the trust information	166
4.20	Possible updates and extensions on the trust matrix	169
4.21	Phase 3: Validation	171
4.22	State diagram illustrating the processing of a write request of the trust information from past experience and by reputation	176

5.1	Sequence structure for Chapter 5	184
5.2	UML component diagram that represents the architectural components of the Trust-Based Access Control Framework	186
5.3	TBAC UML static class diagram	187
5.4	<code>initializePackage</code>	188
5.5	<code>searchPackage</code>	193
5.6	<code>storagePackage</code>	201
5.7	<code>aggregatePackage</code>	205
5.8	Trust Agreements Repository	209
5.9	The module <code>AgreementInterceptor</code>	212
5.10	Resource Description Repository	215
5.11	Resource definition in RDF	216
5.12	The module <code>Service&ResourceInterceptor</code>	217
5.13	The module <code>interactionInterceptor</code>	218
5.14	Identity Repository	219
5.15	Principals data representation in a LDAP Directory	220
5.16	General flow chart representing access control decision with respect to trust information	222
6.1	Sequence structure for Chapter 6	228
6.2	UML application diagram that shows an interaction: Learner–Training Portal Provider	231
6.3	UML application diagram that shows an interaction: Provider–Provider (CoT members)	232
6.4	UML application diagram that shows an interaction: Learners–Mentors/Tutors	233
6.5	UML activity diagram that illustrates the initialization phase	237
6.6	UML Activity diagram for the interaction: Learner–Training Portal Provider	239
6.7	Segment from the UML Activity diagram for the interaction "Learner–Training Portal Provider" highlighting the <code>searchPackgae</code> with regard to the QoS constraints provided by the learner	240
6.8	Segment from the UML Activity diagram for the interaction "Learner–Training Portal Provider" highlighting the <code>aggregationPackage</code>	241
6.9	Segment from the UML Activity diagram for the interaction "Learner–Training Portal Provider" highlighting the <code>Access Decision Engine</code>	242
6.10	Segment from the UML Activity diagram for the interaction "Learner–Training Portal Provider" highlighting the <code>storagePackage</code>	243

6.11 UML Activity diagram for the interaction: Provider–Provider	245
6.12 UML Activity diagram for the interaction: Learners–Mentors/Tutors . .	247
6.13 Exemplary trust behavior of a principal	249
6.14 Influence of malicious ratings	250
6.15 Time performance for the function <code>traverseGraph()</code> of the <code>searchPackage</code>	252
6.16 Interaction between an application and a trust-management system [IK03]	256
6.17 Extension of the KeyNote System with the TBAC Framework	257
7.1 Fulfillment of the requirements in light of the criteria catalogue	265

List of Tables

2.1	Characteristics of static CoTs	31
2.2	eLearning static online community in light of the formal definition of the CoT	36
2.3	Digital Library characteristics in light of the formal definition of the CoT	49
2.4	Grid environment characteristics in light of the formal definition of the CoT	67
3.1	Fulfillment of the requirements through the PKI Models	93
3.2	Fulfillment of the requirements via the Liberty CoT Models	100
3.3	Fulfillment of requirements for interorganizational scenarios	110
3.4	Fulfillment of requirements for privacy management and risk management	111
3.5	Fulfillment of the requirements for Content Quality Trust	116
4.1	Building trust from exemplary performance parameters for PublishFileAction	145
4.2	Trust from past experience (Client)	147
4.3	Trust from past experience (Server)	147
4.4	Fulfillment of requirements of trust from past experiences	149
4.5	Fulfillment of requirements of trust by reputation	157
4.6	Fulfillment of requirements of Content Quality Trust	164
4.7	Fulfillment of requirements of aggregation and final representation of the trust level	167
4.8	Fulfillment of requirements on the storage of the trust information . . .	170
4.9	Fulfillment of the organizational requirements	174
4.10	Fulfillment of the change management requirements	181
5.1	The function <code>initPast()</code>	189
5.2	The function <code>initRep()</code>	191
5.3	The function <code>storeTrustValue()</code>	193
5.4	The function <code>evaluateRequest()</code>	194

5.5	The function <code>traverseGraph()</code>	195
5.6	The function <code>computeTrust()</code>	197
5.7	The function <code>getEdge()</code>	198
5.8	The function <code>evaluateResult()</code>	200
5.9	The function <code>searchScenario()</code>	200
5.10	The function <code>updateTrust()</code>	202
5.11	The function <code>storeUpdatedValue()</code>	204
5.12	The function <code>aggregate()</code>	206
5.13	The function <code>aggregateTwoDimensions()</code>	208
5.14	Fulfillment of the remaining change management requirements	224

Listings

5.1	A code fragment of the function <code>initPast()</code>	190
5.2	A code fragment of the function <code>storeTrustValue()</code>	192
5.3	A code fragment of the function <code>traverseGraph()</code>	195
5.4	A code fragment of the function <code>traverseGraph()</code>	196
5.5	A code fragment of the function <code>computeTrust()</code>	198
5.6	A code section of the function <code>getEdge()</code>	199
5.7	A code section of the function <code>evaluateResults()</code>	199
5.8	A code fragment of the function <code>updateTrust()</code> for updating the trust values from past experience	203
5.9	A code fragment of the function <code>updateTrust()</code> for updating the trust values from the reputation values	204
5.10	A code fragment of the function <code>storeUpdatedValue()</code>	204
5.11	A code fragment of the function <code>aggregate()</code>	206
5.12	A code fragment of the function <code>aggregate()</code>	207
5.13	A code fragment of the function <code>aggregateTwoDimensions()</code>	208
5.14	Provider policy example	213
5.15	Exemplary XSLT for extracting and representing the quality parameters	214
5.16	Exemplary resource definition in RDF	217

List of Algorithms

1	Breadth-first search for requester P_x	139
2	computeTrust: Compute $T_{l_{P_1}P_x}$	151
3	ExternSearch: Estimate $T_{l_{P_1}P_x}$	159
4	Aggregation Algorithm: Estimate $T_{l_{P_x}}^{final}$	163
5	Exemplary trust and risk assessment	178

Bibliography

- [AB00] F. Maghoul A. Broder, R. Kumar. Graph structure in the web. In *Proceedings of the 9th international World Wide Web conference: The international journal of computer and telecommunications networking*, pages 309 – 320, Amsterdam, The Netherlands, February 2000.
- [AD01] Karl Aberer and Zoran Despotovic. Managing trust in a peer-2-peer information system. In Henrique Paques, Ling Liu, and David Grossman, editors, *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM01)*, pages 310–317. ACM Press, 2001.
- [AdA07] B. Thomas Adler and Luca de Alfaro. A content-driven reputation system for the wikipedia. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 261–270, New York, NY, USA, 2007. ACM Press.
- [AEH⁺04] S. R. Amendolia, F. Estrella, W. Hassan, T. Hauer, D. Manset, R. McClatchey, D. Rogulin, and T. Solomonides. Mammogrid: A service oriented architecture based medical grid application, 2004.
- [AM02] F. Azzedin and M. Maheswaran. Evolving and managing trust in grid computing systems. In *Electrical and Computer Engineering Canadian Conference. IEEE CCECE 2002.*, pages 1424–1429, TR Labs, Manitoba Univ., Winnipeg, Man., Canada;, 2002.
- [AR97] Alfarez Abdul-Rahman. The PGP Trust Model. In *Journal of Electronic Commerce*, 1997.
- [ARH00] Alfarez Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6*, page 6007, Washington, DC, USA, 2000. IEEE Computer Society.
- [BD08] L. Boursas and V. Danciu. Dynamic inter-organizational cooperation setup in Circle-of-Trust environments. In *Network Operations and Management Symposium. NOMS 2008. IEEE*, pages 113–120, Salvador, Bahia, Brazil, April 2008.
- [BFIA99] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. RFC 2704: The keynote trust-management system version 2. Technical report, sep 1999.

- [BFIK99] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. The role of trust management in distributed systems security. pages 185–210, 1999.
- [BFL96] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *SP '96: Proceedings of the 1996 IEEE Symposium on Security and Privacy*, page 164, Washington, DC, USA, 1996. IEEE Computer Society.
- [BH06a] L. Boursas and W. Hommel. Efficient Technical and Organizational Measures for Privacy-aware Campus Identity Management and Service Integration. In *12th International Conference of European University Information Systems (EUNIS 2006)*, Tartu, Estonia, Juni 2006.
- [BH06b] L. Boursas and W. Hommel. Policy-based Service Provisioning and Dynamic Trust Management in Identity Federations. In *proceedings of IEEE International Conference on Communications (ICC 2006)*, Istanbul, Turkey, Juni 2006.
- [BH08] L. Boursas and W. Hommel. Propagating Trust and Privacy Aspects in Federated Identity Management Scenarios. In *Proceedings of the 2008 Workshop of HP Software University Association (HP-SUA)*, Marrakech, Morocco, Juni 2008.
- [Bou07] L. Boursas. Virtualization of the Circle of Trust amongst Identity Federations. In *1st International DMTF Academic Alliance Workshop on Systems and Virtualization Management: Standards and New Technologies*, IRIT and Univeristé Paul Sabatier, Toulouse, Frankreich, Oktober 2007.
- [BR07] L. Boursas and H. Reiser. Derivation and use of trust and risk management parameters in dynamic federated environments. In *Proceedings of the 14th Annual Workshop of HP Software University Association*, Leibniz Supercomputing Center, Munich, Germany, Juli 2007.
- [BS02] Sviatoslav Braynov and Tuomas S. Contracting with uncertain level of trust. *Computational Intelligence*, 18:501–514, 2002.
- [BS04] Ezedin Barka and Ravi Sandhu. Role-based delegation model/ hierarchical roles (rbdm1). *acsac*, 0:396–404, 2004.
- [BZ02] Bharat K. Bhargava and Yuhui Zhong. Authorization based on evidence and trust. In *DaWaK 2000: Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, pages 94–103, London, UK, 2002. Springer-Verlag.
- [CCD⁺07] J. G. Cederquist, R. Corin, M. A. C. Dekker, S. Etalle, J. I. den Hartog, and G. Lenzini. Audit-based compliance control. *Int. J. Inf. Secur.*, pages 133–151, 2007.
- [CCT03] Dickson K. W. Chiu, S. C. Cheung, and Sven Till. A three-layer architecture for e-contract enforcement in an e-service environment. In *HICSS '03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 3*, page 74.1, Washington, DC, USA, 2003. IEEE Computer Society.

- [CFL⁺97] Yang-Hua Chu, Joan Feigenbaum, Brian LaMacchia, Paul Resnick, and Martin Strauss. Referee: trust management for web applications. *Comput. Netw. ISDN Syst.*, 29(8-13):953–964, 1997.
- [Cla99] Roger Clarke. Internet privacy concerns confirm the case for intervention. *Commun. ACM*, 42(2):60–67, 1999.
- [CW03] Kari Chopra and William A. Wallace. Trust in electronic environments. *hicss*, 09:331a, 2003.
- [DAC93] Security frameworks in open systems – part 3: Access control. SO/IEC DIS 10181-3 Information technology – Open Systems Interconnection, 1993.
- [DBWS06] Pierpaolo Dondio, Stephen Barrett, Stefan Weber, and Jean Seigneur. Extracting trust from domain analysis: A case study on the wikipedia project. pages 362–373. 2006.
- [DC97] P. DONEY and J. CANNON. An examination of the nature of trust in buyer-seller relationships. *Journal of Marketing* 61, 1997.
- [Del00] Chrysanthos Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *EC'00: Proceedings of the 2nd ACM conference on Electronic commerce*, pages 150–157, New York, NY, USA, Oktober 2000. ACM.
- [DZF03] Li Ding, Lina Zhou, and Tim Finin. Trust based knowledge outsourcing for semantic web agents. In *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, pages 379–387, October 2003.
- [Edi05] T. M. Editor. Oasis extensible access control markup language (xacml) 2.0, core specification. OASIS XACML Technical Committee Standard, 2005.
- [EFL⁺99] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. Spki certificate theory, 1999.
- [Ell] Carl Ellison. Spki/sdsi certificate documentation. <http://world.std.com/cme/html/spki.html>.
- [FBK99] David F. Ferraiolo, John F. Barkley, and D. Richard Kuhn. A role-based access control model and reference implementation within a corporate intranet. *ACM Transactions on Information and System Security*, pages 34–64, 1999.
- [Fer05] Luis Ferreira. *Grid Computing in Research And Education*. IBM Press, 2005.
- [FKNT02] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration, 2002.
- [FPHKH00] Batya Friedman, Jr. Peter H. Khan, and Daniel C. Howe. Trust online. *Commun. ACM*, 43:34–40, 2000.

- [Gal04] Michael Galla. *Social Relationship Management in Internet-based Communication and Shared Information Spaces*. PhD thesis, Institut für Informatik der Technischen Universität München, 2004.
- [GBW⁺98] F. Griffel, M. Boger, H. Weinreich, W. Lamersdorf, M. Merz, and Ponton Hamburg. Electronic contracting with cosmos - how to establish, negotiate and execute electronic contracts on the internet. In *Electronic Contracts on the Internet. 2nd Int. Enterprise Distributed Object Computing Workshop (EDOC '98)*, 1998.
- [GH04] Jennifer Golbeck and James Hendler. Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. In *International Conference on Knowledge Engineering and knowledge Management (EKAW)*, Northamptonshire, 2004.
- [GHP03] J. Golbeck, J. Hendler, and B. Parsia. Trust Networks on the Semantic Web. In *12th International Web Conference (WWW03)*, May 2003.
- [Gil05] Jim Giles. Internet encyclopaedias go head to head. *Nature*, 438(1476-4687 (Electronic)):900–901, 2005.
- [Gol05] Jennifer Ann Golbeck. *Computing and applying trust in web-based social networks*. PhD thesis, Faculty of the Graduate School of the University of Maryland, College Park, 2005.
- [GRI] Gridftp. <http://www.globus.org/grid/software/data/gridftp.php>.
- [GS00] T. Grandison and M. Sloman. A survey of trust in internet application, 2000.
- [Hel05] Burt Helm. Wikipedia: "A Work in Progress". *businessweek*, December 2005.
- [HJS04] T. Dong Huynh, Nicholas R. Jennings, and Nigel R. Shadbolt. Fire: An integrated trust and reputation model for open multi-agent systems. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI)*, pages 18–22, 2004.
- [HJS06] Trung Dong Huynh, Nicholas R. Jennings, and Nigel R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13:119–154, 2006.
- [HMM⁺00] Amir Herzberg, Yosi Mass, Joris Michaeli, Yiftach Ravid, and Dalit Naor. Access control meets public key infrastructure, or: Assigning roles to strangers. In *SP '00: Proceedings of the 2000 IEEE Symposium on Security and Privacy*, page 2, Washington, DC, USA, 2000. IEEE Computer Society.
- [Hof99] Yigal Hoffner. Supporting contract match-making. In *RIDE '99: Proceedings of the Ninth International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises*, page 64, Washington, DC, USA, 1999. IEEE Computer Society.

- [Hom07] W. Hommel. *Architektur- und Werkzeugkonzepte für föderiertes Identitäts-Management*. PhD thesis, Ludwig Maximilians Universität München, July 2007.
- [HR05] Wolfgang Hommel and Helmut Reiser. Federated identity management: Shortcomings of existing standards. In *In 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005) – Managing New Networked Worlds, IEEE*, Nice, France, Mai 2005.
- [ID-04a] Liberty alliance id-ff 1.2 specifications. http://www.projectliberty.org/resource_centers/specifications/liberty_alliance_id_ff_1_2_specifications, May 2004.
- [ID-04b] Liberty alliance id-wsf 2.0 specifications. http://www.projectliberty.org/liberty/resource_centers/specifications/liberty_alliance_id_wsf_2_0_specifications_including_errata_v1_0_updates, May 2004.
- [IK03] John Ioannidis and Angelos D. Keromytis. Experience with the keynote trust management system: Applications and future directions. In *In Proceedings of the 1st International Conference on Trust Management*, pages 284–300. Springer-Verlag, 2003.
- [IPS02] Trust management for ipsec. *ACM Trans. Inf. Syst. Secur.*, 5(2):95–118, 2002.
- [ITU93] ITU-T Rec. X.509 (revised), The Directory - Authentication Framework. International Telecommunication Union, 1993.
- [Jen02] C. D. Jensen. Secure environments for collaboration among ubiquitous roaming entities secure. In *First Internal iTrust Workshop on Trust Management in Dynamic Open Systems*, Glasgow, Scotland, sep 2002.
- [JES00] M. Mastrorocco J. Eller and B. Stauffer. *Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP)*. US-Department of Defense, July 2000.
- [KER] The kerberos network authentication service (v5) (rfc 4120). <http://tools.ietf.org/html/rfc4120>.
- [KFJ01] Lalana Kagal, Tim Finin, and Anupam Joshi. Trust-based security in pervasive computing environments. *IEEE Computer*, December 2001.
- [KGV00] Marjanca Koetsier, Paul W. P. J. Grefen, and Jochem Vonk. Contracts for cross-organizational workflow management. In *EC-WEB '00: Proceedings of the First International Conference on Electronic Commerce and Web Technologies*, pages 110–121, London, UK, 2000. Springer-Verlag.
- [Kle97] Rolf Klein. *Algorithmische Geometrie*. Addison-Wesley, Bonn, 1997.
- [KMW00] Michael Koch, Kathrin Möslin, and Michael Wagner. Vertrauen und reputation in online-anwendungen und virtuellen gemeinschaften. In M. Engelen and D. Neumann, editors, *Proc. Gemeinschaften in neuen Medien (GeNeMe2000)*, pages 69–83, October 2000.

- [KSGm03] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-molina. The eigentrust algorithm for reputation management in p2p networks. In *In Proceedings of the Twelfth International World Wide Web Conference*, pages 640–651. ACM Press, 2003.
- [Lin03] J. Linn. Liberty Trust Models Guidelines V.1.0. Liberty Alliance Specification, 2003.
- [LKDK02] Heiko Ludwig, Alexander Keller, Asit Dan, and Richard King. A service level agreement language for dynamic electronic services. *wecwis*, 00:25, 2002.
- [LLT00] J. J. Longstaff, M. A. Lockyer, and M. G. Thick. A model of accountability, confidentiality and override for healthcare and other applications. In *RBAC '00: Proceedings of the fifth ACM workshop on Role-based access control*, pages 71–76, New York, NY, USA, 2000. ACM.
- [Mar94] S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Sterling, 1994.
- [Max05] E. Michael Maximilien. Agent-based trust model involving multiple qualities. In *In Proc. of the 4th Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 519–526, 2005.
- [MB95] Zoran Milosevic and Andy Bond. Electronic commerce on the internet: What is still missing. In *In Proc. of the 5th Conf. of the Internet Society*, pages 245–254, 1995.
- [MRD05] N. Meyer, A. Rifaut, and E. Dubois. Towards a risk-based security requirements engineering framework. *Workshop on Requirements Engineering for Software Quality. In Proc. of REFSQ'05*, 2005.
- [MS04a] E. Michael Maximilien and Munindar P. Singh. A framework and ontology for dynamic web services selection. *IEEE Internet Computing*, 8(5):84–93, 2004.
- [MS04b] Michael E. Maximilien and Munindar P. Singh. Toward autonomic web services trust and selection. In *ICSOC '04: Proceedings of the 2nd international conference on Service oriented computing*, pages 212–221, New York, NY, USA, 2004. ACM Press.
- [ONT06] Trust Ontology. <http://www.mindswap.org/golbeck/web/trust.daml/>, 2006.
- [Pap08] Elvis Papalilo. *Distributed Trust Management in Grid Computing Environments*. PhD thesis, Fachbereich Mathematik und Informatik, Universität Marburg, Februar 2008.
- [Par07] David Parmenter. *Key Performance Indicators (KPI): Developing, Implementing, and Using Winning KPIs*. 2007.
- [PCM97] R. PETERS, V. COVELLO, and D. MCCALLUM. The determinants of trust and credibility in environmental risk communication: An empirical study. *Risk Analysis* 17, Issue 1, pages 43–54, 1997.

- [PER] Privilege and Role Management Infrastructure Standards (PERMIS). <http://www.permis.org/>.
- [RD02] Matthew Richardson and Pedro Domingos. The intelligent surfer: probabilistic combination of link and content information in pagerank. In *In Advances in Neural Information Processing Systems*, pages 1441–1448. MIT Press, 2002.
- [RDF] Resource description framework (rdf). <http://www.w3.org/RDF/>.
- [SAM03] Security assertion markup language (saml) v2.0. <http://www.oasis-open.org/specs/samlv2.0>, August 2003.
- [SEI] Controversy over Wikipedia’s biography of John Seigenthaler Sr. http://en.wikipedia.org/wiki/Controversy_over_Wikipedia%27s_biography_of_John_Seigenthaler_Sr.
- [Sen98] G. Senizergues. Decidability of bisimulation equivalence for equational graphs of finite out-degree. In *39th Annual Symposium on Foundations of Computer Science*, pages 120 – 129, Nov 1998.
- [SFK] Ravi Sandhu, David Ferraiolo, and Richard Kuhn. The NIST model for role-based access control: Towards a unified standard. pages 47–64.
- [SFR00] Michael Schillo, Petra Funk, and Michael Rovatsos. Using trust for detecting deceitful agents in artificial societies. In *Applied Artificial Intelligence, Special Issue on Trust, Deception and Fraud in Agent Societies*, 14(8):825–848, September 2000.
- [SLA05] R. Gandhi S. Lee and G. Ahn. Security requirements driven risk assessment for critical infrastructure information systems. In *SREIS’05*, 2005.
- [Slo04] Morris Sloman. Trust Management in Internet and Pervasive Systems. *IEEE Intelligent Systems*, 19(5):77–79, September 2004.
- [Ste06] Randy A. Steinberg. *Measuring ITIL: Measuring, Reporting and Modeling - the IT Service Management Metrics That Matter Most to IT Senior Executives*. Trafford Publishing, August 2006.
- [THCS01] Ronald L. Rivest Thomas H. Cormen, Charles E. Leiserson and Clifford Stein. Introduction to algorithms. pages 531–539. MIT Press and McGraw-Hill, 2001.
- [TSY94] Roshan K. Thomas, Ravi S. S, and Hu Y. Conceptual foundations for a model of task-based authorizations. In *Computer Security Foundations Workshop VII, CSFW 7. Proceedings*, pages 66–79, 1994.
- [VN97] Nalini Venkatasubramanian and Klara Nahrstedt. An integrated metric for video qos. In *MULTIMEDIA ’97: Proceedings of the fifth ACM international conference on Multimedia*, pages 371–380, New York, NY, USA, 1997. ACM.
- [VSH05] Hogan Victoria Sheckler and Hartson. Liberty Alliance Contractual Framework Outline for Circles of Trust. Liberty Alliance Specification, 2005.

- [Wag06] Hoda Waguih. A proposed trust model for the semantic web. In *PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY*, February 2006.
- [Was04] Thomas Wason. Liberty id-ff architecture overview v1.2. Liberty Alliance Specification, 2004.
- [WST04] Web services trust language (ws-trust). <http://www.ibm.com/developerworks/library/specification/ws-trust/>, May 2004.
- [WYS⁺02] Marianne Winslett, Ting Yu, Kent E. Seamons, Adam Hess, Jared Jacobson, Ryan Jarvis, Bryan Smith, and Lina Yu. Negotiating trust on the web. *IEEE Internet Computing*, 6:30–37, 2002.
- [X50] Internet X.509 Public Key Infrastructure Certificate and CRL Profile. <http://www.ietf.org/rfc/rfc2459.txt>.
- [XKM] Xml key management specification (xkms). <http://www.w3.org/TR/xkms/>.
- [XL02] Li Xiong and Ling Liu. Building trust in decentralized peer-to-peer electronic communities. In *The 5th International Conference on Electronic Commerce Research (ICECR)*, 2002.
- [XML] Xml encryption wg. <http://www.w3.org/Encryption/>.
- [YS02] Bin Yu and Munindar P. Singh. An evidential model of distributed reputation management. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 294–301, New York, NY, USA, 2002. ACM.
- [YW03] Ting Yu and Marianne Winslett. Policy migration for sensitive credentials in trust negotiation. In *WPES '03: Proceedings of the 2003 ACM workshop on Privacy in the electronic society*, pages 9–20, New York, NY, USA, 2003. ACM.
- [ZAD⁺06] Honglei Zeng, Maher Alhossaini, Li Ding, Richard Fikes, and Deborah L. McGuinness. Computing trust from revision history. In *Proceedings of the 2006 International Conference on Privacy, Security and Trust*, October 2006.
- [Zim94] P. Zimmermann. PGP User's guide. In *MIT Press*, 1994.
- [ZM00] Giorgos Zacharia and Pattie Maes. Trust management through reputation mechanisms. volume 14, pages 881–907, 2000.
- [ZMM99] Giorgos Zacharia, Alexandros Moukas, and Pattie Maes. Collaborative reputation mechanisms in electronic marketplaces. In *HICSS '99: Proceedings of the Thirty-second Annual Hawaii International Conference on System Sciences-Volume 8*, page 8026, Washington, DC, USA, 1999. IEEE Computer Society.