TUM

TECHNISCHE UNIVERSITÄT MÜNCHEN
INSTITUT FÜR INFORMATIK

Data Usage Management on the Web

Lalana Kagal and Alexander Pretschner

TUM-I1212

Lalana Kagal and Alexander Pretschner (eds):

Proceedings of the WWW2012 workshop on

Data Usage Management on the Web

Lyon, April 16th, 2012

Preface

More and more data is being made available for use by online services or the public at large such as data managed by governments and/or companies. In addition, users are constantly disclosing information to services such as search engines, social networks, and media hosting sites in exchange for better and more personalized results, community sharing, and social and professional interaction. As public and private data flows through services that benefit users, it becomes harder to control how much of this data is stored or how it is used and shared by these different services.

Data usage control generalizes access control in order to understand what happens to data after it has been given away (accessed). Spanning the domains of privacy, the protection of intellectual property and compliance, typical current requirements include "delete after thirty days", "don't delete within five years", "notify whenever data is given away", and "don't print". However, in the near future, more general requirements may include "do not use for employment purposes", "do not use for tracking", as well as "do not use to harm me in any way". Major challenges in this field include policy specification, the relationship between end user actions and technical events, tracking data across layers of abstraction and logical as well as physical systems, policy enforcement, and policy guarantees.

The goal of the WWW 2012 workshop of Data Usage Management on the Web (http://dig.csail.mit.edu/2012/WWW-DUMW/) was to discuss current technical developments in usage control and, in particular, foster collaboration in the area of usage representation, provenance tracking, misuse identification, and distributed usage enforcement. Though enabling privacy through careful and controlled dissemination of sensitive information is closely related to usage control, this workshop was more broadly interested in understanding data usage control as a whole.

The workshop papers in these proceedings discuss the state of the art in different approaches including preventive (such as Digital Rights Management systems) and forensic (such as accountability) approaches, identify open problems, and provide innovative solutions to some problems of data usage control.
The workshop also included a keynote by Prof. Ravi Sandhu of University of Texas at San Antonio that discussed some of the tough and grand challenges in this area. He defined data usage control as unifying a number of privacy, confidentiality and intellectual property protection requirements that have been present in the literature since the earliest days of cyber security. He suggested that usage control is concerned with handling of data before, during and after access, and it has been widely practiced on the Web at least in rudimentary form. Prof. Sandhu concluded by postulating that in the future, usage control would require additional sophistication in models as well as in technical and non-technical enforcement.

While data usage control in restricted contexts already is a difficult problem, the workshop has shown that data usage control in open environments such as the Web provides a plethora of technical, social, and ethical challenges.

This workshop is the third in a series of related workshops: a Dagstuhl seminar on Distributed Usage Control in April 2010 and a W3C workshop on Privacy and Data Usage Control in October 2010.

Boston, Munich, August 2012
Lalana Kagal and Alexander Pretschner

Table of Contents

Keynote Address

Ravi Sandhu: Grand Challenges in Data Usage Control

This talk will give a personal perspective on data usage control models and mechanisms. The concept of data usage control has been formally articulated only relatively recently. It unifies a number of privacy, confidentiality and intellectual property protection requirements that have been present in the literature since the earliest days of cyber security. In my perspective, usage control is concerned with handling of data before, during and after access, and it has been widely practiced on the web at least in rudimentary form. In future it will require additional sophistication in models as well as in technical and non-technical enforcement. The talk will speculate on some of the tough and grand challenges in this arena.

A Critical Look at Decentralized Personal Data Architectures

Arvind Narayanan
relax@stanford.edu

Solon Barocas
solon@nyu.edu

Vincent Toubiana
vincent.toubiana@alcatel-lucent.com

Helen Nissenbaum
hfn1@nyu.edu

Dan Boneh
dabo@cs.stanford.edu

ABSTRACT

While the Internet was conceived as a decentralized network, the most widely used web applications today tend toward centralization. Control increasingly rests with centralized service providers who, as a consequence, have also amassed unprecedented amounts of data about the behaviors and personalities of individuals.

Developers, regulators, and consumer advocates have looked to alternative decentralized architectures as the natural response to threats posed by these centralized services. The result has been a great variety of solutions that include personal data stores (PDS), infomediaries, Vendor Relationship Management (VRM) systems, and federated and distributed social networks. And yet, for all these efforts, decentralized personal data architectures have seen little adoption.

This position paper attempts to account for these failures, challenging the accepted wisdom in the web community on the feasibility and desirability of these approaches. We start with a historical discussion of the development of various categories of decentralized personal data architectures. Then we survey the main ideas to illustrate the common themes among these efforts. We tease apart the design characteristics of these systems from the social values that they (are intended to) promote. We use this understanding to point out numerous drawbacks of the decentralization paradigm, some inherent and others incidental. We end with recommendations for designers of these systems for working towards goals that are achievable, but perhaps more limited in scope and ambition.

1. BRIEF HISTORICAL OVERVIEW

The search for alternatives to centralized aggregation of personal data began in the late 1990s which saw a wave of so-called 'negotiated privacy techniques' including commercial 'infomediaries' [24, 16]. These entities would store consumers' data and help facilitate the drafting of contracts that set the terms of the exchange and use of data. The 1999 book *Net Worth* [23] galvanized both industry and privacy advocates, generating hopes for a future in which privacy problems could be solved through a mix of decentralized storage and private contracts, potentially obviating the need for privacy law or even the adoption of fair information practices [10, 60].

Within five years, nearly all of this excitement had faded and all commercial (Persona, Privada, Lumeria, etc.) and community (P3P) initiatives had floundered [1] — some in truly spectacular fashion, such as AllAdvantage. And yet, by the end of the decade, many new initiatives and projects that shared almost identical goals emerged. Vendor Relationship Management (VRM) [35] has gained steady momentum as a general set of principles that aim simultaneously to improve user privacy, enhance customer autonomy, and increase market efficiency through a combination of mechanisms that aggregate data in a single (per-user) repository under users' control and tools to negotiate agreements that would grant outside organizations access to and use of that data.

Parallel efforts to develop so-called personal data stores (PDS), personal data servers, personal data lockers/vaults, and personal clouds [18] have focused more narrowly on the platforms and protocols to support unified repositories of user data that could be managed locally by the user or outsourced to a trusted third party. The impetus for these projects are varied, ranging from user interest in aggregating one's own data in a single location to better derive benefits from their mixing and matching to more explicit interests in privacy (user control) and commerce (a market place for sharing, including possibilities for cash payments in exchange for data) [13].

The similarities between these and earlier efforts can be quite stark: Mydex's recent white paper, "The Case for Personal Information Empowerment" [38], recapitulates much that was described in a white paper released a full decade earlier by Lumeria, a failed infomediary [30]. To describe this as a simple case of "an idea whose time has come" would be to miss the important lessons that these earlier and recurring failures should offer those who wish to pursue decentralized personal data architectures.

Decentralized social networking has been a largely parallel, sometimes overlapping line of development with similar motivations. We subdivide such social networks into federated (ecosystem of interoperable implementations in the client-server model) and distributed (peer-to-peer). The term distributed social networking is frequently but incorrectly used to describe all decentralized social networks.

While some early thinking in the semantic web community could be classified in this category,[1] for the most part decentralized social networking appears not to have anticipated the success of mainstream commercial, centralized social networks, but rather developed as a response to it. Indeed, prominent members of the web community dismissed social networks until 2007–2008 (for example, [27] and [15]) and academic computer scientists appear to have considered it a passing fad as well — in our survey we see a sharp spike in interest among researchers around this time frame.

A series of well-publicized privacy mishaps by Facebook and Google starting in 2009 that reached its crescendo around the 2010 f8 developer conference stirred up interest among the public and policymakers.[2] Perhaps the most well known project that resulted is Diaspora[3], which was funded in excess of $200,000 via the crowd funding platform kickstarter.com. As of this writing Wikipedia lists about 40 decentralized social networks [58], most of which are federated, whereas the academic literature has focused on distributed social networking for natural reasons, since those present more research challenges.

2. REPRESENTATIVE SURVEY

Rather than attempt an exhaustive survey, in this section we list the key ideas that have been explored in the course of developing decentralized designs. There has been a great fecundity of creative and complex ideas in this space spanning the realms of technology, law and economics; we are unable to present them in detail due to space constraints. We refer the reader to the cited works.

The core idea of an infomediary is that of a trusted third party that interfaces between the user and commercial entities such as marketers [23]. Users' personal data can be manually given to the infomediary, as in Lumeria, or collected through passive monitoring, as in AllAdvantage and other systems [20]. That information can then be utilized without explicit monetization (Mydex, etc.), or users can be paid for their data (AllAdvantage, Bynamite [29], etc). It has variously been argued that telecommunications providers [55, 4], banks [9] and other parties such as providers of home entertainment set-top boxes are ideally suited to play the role of the intermediary. An infomediary might also enable a targeted *attention market* [39] based on user preferences.

Kang et al. introduce the intriguing idea of *licensing* intermediaries to increase their trustworthiness [28]. In the other direction, Vendor Relationship Management systems largely eliminate the infomediary as a separate entity, and instead replace it with a software agent [35]. Some software intermediaries like Adnostic use cryptography to achieve additional privacy properties [54]. Other ideas for improving privacy include fine-grained access control lists [37].

Both VRM and infomediary systems often emphasize benefits to the firm from the intermediated nature of the exchange. Goldman [21] envisions that software agents will make marketing messages perfectly relevant, eliminating externalities from wasted attention. By Coase's theorem [34], this will lead to a socially optimal level of marketing.

Turning to social networks, the key challenge of distributed social networks is hosting and message transfer. One solution is to encrypt messages and store them in a distributed hash table [8, 2]. Another is "social replication": messages are stored in plaintext in a redundant manner by those who have access rights (typically friends of the message poster) [49]. Message passing sometimes exploits the relationship between the social graph and the topology of the physical network [25, 8].

Another frequent goal is keeping edges of the graph secret, for which various solutions have been proposed: a cryptographic approach [5], anonymous routing [14] and friend-to-friend networks such as Freenet in 'darknet' mode [12]. Persona takes the cryptographic heavy-lifting a step further to enable fine-grained access control using attribute-based encryption [6].

Other models for hosting have been explored. In vis-a-vis, each user owns an EC2 virtual host that is active at all times [48], whereas FreedomBox[4] proposes cheap plug computers. Lam et al. have proposed email as a backend [19] and ephemeral networks on smartphones [17]. Unhosted[5] proposes separating data from code, but keeping both in the cloud. Along similar lines, Frenzy[6] is a distributed social network software with Dropbox as the backend. Polaris proposes reducing existing social networks such as Youtube and Twitter to datastores and layering a social network on top, with smartphones providing access control management interfaces [59].

Finally, federated social networks aim to create an ecosystem of standards-based interoperable implementations of social networks. Some designs such as Diaspora are a hybrid between distributed and federated. OStatus, being coordinated by the W3C, represents an interesting approach to standardization for federated microblogging: it references a suite of existing protocols rather than developing them from scratch.

3. CLASSIFICATION

Table 1: The four types of architectures that are the subject of our study

	Commerce, Health etc.	Social Networking
Self-hosted	PDS / VRM	Distributed
Outsourced	Infomediary	Federated

We emphasize that the division in Table 1 is only meant to

[1] The Internet Archive lists a version of the *Friend of a Friend* (FOAF) project (www.foaf-project.org) from August 2003, and other efforts may be older.

[2] For an article typifying public opinion during that period, see [45].

[3] `https://joindiaspora.com/`

[4] `http://freedomboxfoundation.org/`

[5] `http://unhosted.org/`

[6] `http://frenzyapp.com/`

provide the reader with a rough mental map and is far from precise. The vertical axis, in particular, is closer to a spectrum than a strict division. The terms Personal Data Store and Vendor Relationship Management do not appear to have a single definition. Also, some PDS projects are application-agnostic, but these tend to be software libraries/platforms rather than complete user-facing systems.

Towards a finer-grained classification and understanding of different projects, we propose the following (non-independent) axes that are components of what it means for an architecture to be decentralized.

1. Locus of data hosting: this could be remote (centralized), by a trusted third party (infomediary), distributed (peer-to-peer), or local (i.e., on the user's device).
2. Open standards vs. proprietary.
3. Open vs. closed-source implementations.
4. Data portability: Data export (for users), APIs (for third parties), or none.

The above are technical characteristics; one might also try to classify systems in terms of the social values they espouse. We discuss four in particular.

1. Privacy: According to Nissenbaum [41, 40], systems that attempt to preserve privacy should attempt to preserve the integrity of the context in which actors engage with each other. They should do this by ensuring that information flows respect the norms of the context. To the degree that systems better model and mediate appropriate information flows, they will advance the privacy interests of their users. This view will inform the discussion in Section 4.1.
2. Utility: We refer to the overall social benefit of the system, in the sense of welfare maximiation in economics. One way to achieve increased utility is through greater interoperability or data portability.
3. Cost: Cost encompasses hosting and bandwidth costs as well as software development and maintenance costs. Centralized and decentralized systems behave very differently: in the former case there is typically a single entity that bears all the costs whereas in the decentralized setting it can be split among users and various software creators and service providers. Comparing these alternatives may therefore be tricky.
4. Innovation: We must also consider how quickly different systems are able to evolve and adapt. Some have argued that open standards catalyze innovation while others point to the time and monetary costs of standardization. The strength of the business model, the extent of market competition, the ability to harness and analyze data, and legal compliance requirements are some of the other factors that affect how conducive a system is to innovation.

Values may not be immediately deducible from the technical design of a system, but may instead only be observable empirically. Indeed, we suggest that much of the reason for what we see as overenthusiastic claims about decentralized systems is that design characteristics have been confused with values. We discuss two prominent cases in detail in Sections 4.1 and 4.2. Moreover, we doubt whether any architecture could optimize for all values simultaneously.

4. DRAWBACKS OF DECENTRALIZATION

In this section we present some underappreciated drawbacks of decentralized architectures. Not all of these apply to all types of systems, nor is any of them individually a decisive factor. But collectively they may help explain why decentralization faces a steep road ahead, and why even if adopted, decentralization will not necessarily provide all the benefits that its proponents believe will automatically flow from it.

An architecture without a single point of data aggregation, management and control has several **technical** disadvantages. First is functionality: there are several types of computations that are hard or impossible without a unified view of the data. Detection of fraud and spam, search, collaborative filtering, identification of trending topics and other types of analytics are all examples. Decentralized systems also suffer from inherently higher network unreliability, resulting in a tradeoff between consistency and availability (formalized as the CAP theorem [57]); they may also be slower from the user's point of view.[7] The need for synchronized clocks and minimizing data duplication are other challenges.

The benefits and costs of standardization are a prominent socio-technical factor. Many decentralized systems depend on multiple interoperating pieces of software, which requires standardization of technical protocols, design decisions, etc. On the one hand, such an ecosystem could promote long-term innovation; on the other hand, these processes (e.g., HTML5) move at a far slower pace than Facebook or an ad network which can roll out features over the timespan of days or weeks. Shapiro notes two benefits of standardization: greater realization of network effects and protection of buyers from stranding, and one cost: constraints on variety and innovation, and argues that the impact on competition can be either a benefit or a cost [50].

Let us now turn to **economics**. Centralized systems have significant *economies of scale* which encompasses hosting costs, development costs and maintenance costs (e.g., combating malware and spam),[8] branding and advertising [42]. A related point in the context of social networks: we hypothesize that the network effect is stronger for centralized systems due to tighter integration.

Path dependence is another key economic issue: even if we assume that centralized and decentralized architectures represent equally viable equilibria, which one is actually reached might be entirely a consequence of historical accident. Most of the systems under our purview – unlike, say, email – were initially envisioned as commercial applications operating un-

[7] Google reports that users exposed to an additional delay of as little as 100ms performed a statistically significantly smaller number of searches [44].

[8] Facebook has built a highly sophisticated real time "immune system" which relies in part on human operators [51].

der central control, and it is unsurprising they have stayed that way.

The theory of *unraveling* suggests that infomediaries in particular might *not* in fact represent a stable equilibrium. For an infomediary to succeed, consumers and businesses must transact through the intermediary rather than directly with each other. But either side of this market might see participants iteratively defecting, resulting in unraveling of the market. Chen et al. discuss how this might happen from the businesses' side [11], and Peppet discusses it from the consumer side [43]. However, it is not fully clear why many types of intermediaries have taken hold in many other markets — employment agents, goods appraisers, etc — but not in the market for personal data.

A variety of **cognitive** factors hinder adoption of decentralized systems as well. First, the fact that decentralized systems typically require software installation is a significant barrier. Second, more control over personal data almost inevitably translates to more decisions, which leads to cognitive overload. Third, since users lack expertise in software configuration, security vulnerabilities may result. A related point is that users may be unable to meaningfully verify privacy guarantees provided through cryptography.

Finally, we find that decentralized social networking systems in particular fare poorly in terms of mapping the norms of information flow. Access control provides a very limited conception of privacy. We provide several examples. First is the idea of "degrees of publicness." For example, on Facebook a post may be publicly visible, yet the site has defenses to stop crawlers which prevents the post ending up in a search engine cache, so that the user may meaningfully hide or delete the post later if they so choose. Second, in current social networks privacy is achieved not only through technical defenses but also through "nudges" [36]. When there are multiple software implementations, users cannot rely on their friends' software providing these nudges. Third, distributed social networks reveal users' content consumption to their peers who host the content[9] (unless they have a "push" architecture where users always download accessible content, whether they view it or not, which is highly inefficient.) Finally, decentralized social networks make reputation management and "privacy through obscurity" (in the sense of [26]) harder, due to factors such as the difficulty of preventing public, federated data from showing up in search results.

4.1 On Control over Personal Data

We now discuss two drawbacks in detail to illustrate the difference between architectural decisions and social values that they are often implicitly assumed to promote. The first is the distinction between control over hosting and privacy. To elucidate this we present a thought experiment.

What does it mean for users to truly host and control their personal data, while still being able to participate in activities such as social networking and personalized commerce? Compared to using Facebook, hosting one's data on a personal EC2 instance certainly puts the user in greater control, but Amazon will turn over user data in response to a subpoena or court order [3].

For any hope of absolute control, users must, at a minimum, host data on their own device resident on their physical property. This is already considerably at odds with the reality of today's consumer Internet: bandwidth to the home is often asymmetric, or connectivity is restricted in other ways (NATs, firewalls), and few individuals possess always-on devices capable of running web services.[10]

Furthermore, the software running the services must be open-source, and be audited by third-party certification authorities, or by "the crowd". Silent auto-updates, which is the model that client-side software is increasingly moving towards, would be difficult due to the auditing requirement, perhaps prohibitively so.

Further still, *hardware* might have backdoors, and therefore needs an independent trust mechanism as well. The user also needs the time and knowhow to configure redundant backups, manage software security, etc. Finally almost all decentralized architectures face the the problem of "downstream abuse" which is that the user has no technical means to exercise control over use and retransmission of data once it has been shared [47].

This thought experiment shows that absolute control is impossible in practice. Further, it suggests that control over information is probably not the right conceptualization of privacy, if privacy is the end goal.

4.2 Open standards and Interoperability

Interoperability is a laudable goal; it could enhance social utility, as we have mentioned earlier. However, it has frequently been reduced to the notion of open standards. We argue here that while open standards are a prerequisite for interoperability, there is a big gap between the two. In particular, the efforts at federated social networking all follow open standards, but their actual interoperability status in practice appears to be poor [56]. Let us examine why this is the case.

One major impediment is that there are too many standards to choose from. For the most basic, foundational component — identity — there are many choices: OpenID, WebID and others. While it is possible to connect these to each other, it requires extra effort. As we get to more complex (but still basic) functionality such as federation of messages, we find on the one hand Atom/PubSubHubbub etc. and the OStatus suite[11] on top of it, and on the other hand XMPP and the Wave federation protocol[12] on top of it. It appears that the former is gradually winning out, but this is a slow process.

The second major impediment is that as soon as we get past the basics like identity, friendship and status updates, there

[9] This is a particularly serious problem for systems like Contrail [52].

[10] It remains to be seen if smartphones will become practical for this use-case.

[11] `http://ostatus.org/`

[12] `http://www.waveprotocol.org/`

is an incredible array of parameters to nail down. Take the apparently trivial issue of what formatting is allowed in a status update. Unless two systems agree on the same standard, they are not interoperable because users of one will see malformatted messages originating from the other. Needless to say, centralized platforms have a large and ever-increasing set of features — photos, video chat, polls, to name a few — all of which would require standardization in the federated context. Finally, access control in a federated setting presents hard technical challenges.

The practical upshot is that the only suite of standards that shows any signs of meaningful interoperability is StatusNet[13] — microblogging is both text based, largely eliminating the formatting issue, and typically public, sidestepping the access-control issue — although identi.ca remains the only implementation with meaningful adoption. Even though this system limits status updates to text, a version of the formatting problem still plagues it: identi.ca restricts updates to 140 characters in an attempt to maintain some interoperability with Twitter!

We conclude that while federated social networks have the potential to converge on a reasonably interoperable collection of software — subject to the caveats of differing feature sets and parameters — it is not simply a matter of making some technical decisions, but instead needs serious developer commitment as well as the involvement of standards bodies with significant authority.

5. RECOMMENDATIONS

Based on our analysis above, we offer the following recommendations for developers of decentralized systems.

1. Consider the economic feasibility of your design. In particular, are there entities with the economic incentive to play the various roles that are called for? This has perhaps been the most common reason for the lack of adoption of past proposals and projects.

2. Pay heed to conceptual fidelity. Are you shooting at the right target? Do people have the values you think they do? Do they really want the features/benefits you claim they want? As one example, there have been a multiple of projects that attempt encrypted communication over Facebook and other social networks (NOYB [22], FlyByNight [32], Lockr [53], FaceCloak [33], Scramble! [7], etc.), but the lack of adoption suggests that the usability costs do not outweigh the benefits to users.

3. Incorporate other notions of regulability [61, 31]. Many decentralized systems represent an extreme choice: they seek to achieve privacy and other properties purely through technology, ignoring socio-legal approaches. This extreme may not be optimal.

4. Offer advantages other than privacy to users. Privacy is always a secondary feature — while it might tip the balance between two competing products, users rarely pick a product based on privacy alone. For example, distributed social networking can enable some location-specific functionalities through peer-to-peer networking even when there is no Internet access.

5. Design with standardization in mind. One of the disadvantages we have identified is the proliferation of non-interoperable systems. Open standards are not enough: developers must actively prioritize interoperability and write and maintain glue code to interface with other systems.

6. Target limited feature sets. A system like Facebook is a large, complex moving target. Attempting to create a decentralized version of it is a futile endeavor. Instead, systems that embody the 'minimum viable product' strategy might succeed better in the market. Decentralized microblogging appears to be a relatively attainable goal at the present time, and censorship resistance is a goal for which there is much demand.

7. Work with regulators. As numerous law/economics scholars have pointed out, market solutions appear to underprovide privacy and regulation can help tweak the environment to address this imbalance [46]. Those who wish to see the personal data ecosystem flourish would do well to support regulatory interventions such as transparency and opt-out that can help level the playing field between centralized and decentralized systems.

6. CONCLUSION

In this position paper we have taken a look back at the efforts to build decentralized personal data architectures motivated either by discontent with the status quo, or as a better way to organize information markets and leverage new commercial opportunities, or a combination of both. We hope we have provided some mental clarity to the reader on the similarities, differences and common themes between the various systems and brought fresh perspective to the question of why they have largely floundered.

We hope to kick off a more tempered discussion of the future of personal data architectures in both scholarly and hobbyist/entrepreneurial circles, one that is informed by the lessons of history. There is much work to be done along these lines — application of economic theory can shed light on questions such as the relative strength of network effects in centralized vs. decentralized systems. Empirical methodology such as user and developer interviews would also be tremendously valuable. While we have provided some suggestions for developers, in the future we hope to identify specific application domains that are relatively amenable to the adoption of decentralized architectures, as well as to provide concrete recommendations for policymakers who might wish to foster a different market equilibrium.

Acknowledgement. The first author would like to thank Monica Lam and the other members of the MobiSocial project for enlightening discussions, Deirdre Mulligan, Nick Doty and Jennifer King for helping develop ideas on a multi-factor approach to privacy, Alejandro Molnar for general education about economics, Alessandro Acquisti for sharing his bibliography on the economics of privacy and numerous online commenters for perspectives, links and information.

[13] http://status.net/

7. REFERENCES

[1] M. S. Ackerman. The intellectual challenge of CSCW: the gap between social requirements and technical feasibility. *Hum.-Comput. Interact.*, 15(2):179–203, Sept. 2000.

[2] L. M. Aiello and G. Ruffo. Lotusnet: tunable privacy for distributed online social network services. *Computer Communications*, In Press„ 2010.

[3] Amazon, Inc. AWS Customer Agreement. `http://aws.amazon.com/agreement/`.

[4] I. Ayres and M. Funk. Marketing Privacy: A Solution for the Blight of Telemarketing (and Spam and Junk Mail). *SSRN eLibrary*, 2002.

[5] M. Backes, M. Maffei, and K. Pecina. A security api for distributed social networks. In *NDSS*, 2011.

[6] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: An online social network with user-defined privacy. *Computer*, 39(4):135–146, 2009.

[7] F. Beato, M. Kohlweiss, and K. Wouters. Scramble! Your Social Network Data. In *PETS*, pages 211–225, 2011.

[8] S. Buchegger, D. Schioberg, L. H. Vu, and A. Datta. PeerSoN: P2P Social Networking - Early Experiences and Insights. In *Proceedings of the Second ACM Workshop on Social Network Systems Social Network Systems 2009, co-located with Eurosys 2009*, Nurnberg, Germany, March 31 2009.

[9] Carol Matlack. Who Do You Trust More with Your Data: Facebook or a Bank? BusinessWeek, 2012.

[10] J. Catlett. Panel on infomediaries and negotiated privacy techniques. In *Proceedings of the tenth conference on Computers, freedom and privacy: challenging the assumptions*, CFP '00, pages 155–156, New York, NY, USA, 2000. ACM.

[11] Y. Chen, G. Iyer, and P. V. Padmanabhan. Referral Infomediaries and Retail Competition. *SSRN eLibrary*, 2001.

[12] I. Clarke, S. G. Miller, T. W. Hong, O. Sandberg, and B. Wiley. Protecting free with freenet. *Internet Computing IEEE*, 6(February):40–49, 2002.

[13] P. D. E. Consortium. The Startup Circle. `http://personaldataecosystem.org/2011/06/startup/`, 2011.

[14] L. Cutillo, R. Molva, and T. Strufe. Safebook: A privacy-preserving online social network leveraging on real-life trust. *IEEE Communications Magazine*, 47(12):94–101, 2009.

[15] Dave Winer. Why Facebook Sucks. `http://scripting.com/stories/2007/10/13/whyFacebookSucks.html`, 2007.

[16] A. Dix. Infomediaries and negotiated privacy techniques. In *Proceedings of the tenth conference on Computers, freedom and privacy: challenging the assumptions*, CFP '00, pages 167–, New York, NY, USA, 2000. ACM.

[17] B. Dodson and M. Lam. Musubi: A mobile privacy-honoring social network. 2010.

[18] E. Drummond. The Personal Cloud. `http://equalsdrummond.name/2011/02/07/the-personal-cloud/`, 2011.

[19] M. H. Fischer and M. S. Lam. Email Clients as Decentralized Social Apps in Mr . Privacy. In *Proceedings of the fourth Hot Topics in Privacy Enhancing Technologies , co-located with Eurosys 2009*, July 2011.

[20] B. Givens. Infomediaries and negotiated privacy: resources. In *Proceedings of the tenth conference on Computers, freedom and privacy: challenging the assumptions*, CFP '00, pages 165–166, New York, NY, USA, 2000. ACM.

[21] E. Goldman. A Coasean Analysis of Marketing. *Santa Clara Univ. Legal Studies Research Paper No. 06-03.*

[22] S. Guha, K. Tang, and P. Francis. Noyb: privacy in online social networks. In *Proceedings of the first workshop on Online social networks*, WOSN '08, pages 49–54, New York, NY, USA, 2008. ACM.

[23] J. Hagel and M. Singer. *Net worth: shaping markets when customers make the rules.* Harvard Business School Press, 1999.

[24] J. Hagel, III and J. F. Rayport. The coming battle for customer information. In *Creating value in the network economy*, pages 159–171. Harvard Business School Press, Boston, MA, USA, 1999.

[25] L. Han, B. Nath, L. Iftode, and S. Muthukrishnan. Social butterfly: Social caches for distributed social networks. In *SocialCom/PASSAT*, pages 81–86, 2011.

[26] W. Hartzog and F. D. Stutzman. The Case for Online Obscurity. *SSRN eLibrary*, 2010.

[27] Jeff Atwood. Avoiding Walled Gardens on the Internet. `http://www.codinghorror.com/blog/2007/06/avoiding-walled-gardens-on-the-internet.html`, 2007.

[28] J. Kang, K. Shilton, D. Estrin, J. Burke, and M. Hansen. Self-surveillance privacy. *Iowa Law Review*, 97:809, December 2010.

[29] Kashmir Hill. Names You Need To Know in 2011: Bynamite. Forbes, 2011.

[30] F. Labalme and J. Duwaik. An infomediary approach to the privacy problem. *Feb*, 9:24, 1999.

[31] L. Lessig. *Code and other laws of cyberspace.* Basic Books, 1999.

[32] M. M. Lucas and N. Borisov. FlyByNight: mitigating the privacy risks of social networking. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, WPES '08, pages 1–8, New York, NY, USA, 2008. ACM.

[33] W. Luo, Q. Xie, and U. Hengartner. FaceCloak: An Architecture for User Privacy on Social Networking Sites. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 03*, pages 26–33, Washington, DC, USA, 2009. IEEE Computer Society.

[34] A. Marciano. Ronald Coase, 'The Problem of Social Cost' and The Coase Theorem: An anniversary celebration. *European Journal of Law and Economics*, 31(1):1–9, 2010.

[35] A. Mitchell, I. Henderson, and D. Searls. Reinventing direct marketing — with vrm inside. *Journal of Direct Data and Digital Marketing Practice*, 10(1):3–15, 2008.

[36] D. Mori. Privacy nudges protect information. `http://thetartan.org/2010/3/22/scitech/privacynudges`,

2010.

[37] M. Mun, S. Hao, N. Mishra, K. Shilton, J. Burke, D. Estrin, M. Hansen, and R. Govindan. Personal Data Vaults: a locus of control for personal data streams. 2010.

[38] Mydex. The case for personal information empowerment : The rise of the personal data store, 2010.

[39] NASDAQ/Edgar Online. AllAdvantage IPO filing. http://ipo.nasdaq.com/TextSection.asp?cikid=71762&fnid=3262&sec=bd, 1999.

[40] H. Nissenbaum. *Privacy in Context - Technology, Policy, and the Integrity of Social Life*. Stanford University Press, 2010.

[41] H. F. Nissenbaum. Privacy as Contextual Integrity. *SSRN eLibrary*.

[42] Y. Peles. Economies of scale in advertising beer and cigarettes. *The Journal of Business*, 44(1):32–37, 1971.

[43] S. R. Peppet. Unraveling Privacy: The Personal Prospectus and the Threat of a Full Disclosure Future. *Northwestern University Law Review, 2011*, 2010.

[44] G. Research. Speed matters. http://googleresearch.blogspot.com/2009/06/speed-matters.html, 2009.

[45] Ryan Singel. Facebook's Gone Rogue; It's Time for an Open Alternative. http://www.wired.com/epicenter/2010/05/facebook-rogue/, 2010.

[46] P. M. Schwartz. Privacy and Democracy in Cyberspace. *SSRN eLibrary*.

[47] P. M. Schwartz. Property, Privacy, and Personal Data. *SSRN eLibrary*.

[48] A. Shakimov, H. Lim, K. Li, D. Liu, and A. Varshavsky. *Vis-a-Vis: Privacy-Preserving Online Social Networking via Virtual Individual Servers*. IEEE, 2011.

[49] A. Shakimov, A. Varshavsky, L. P. Cox, and R. Cáceres. Privacy, cost, and availability tradeoffs in decentralized osns. In *Proceedings of the 2nd ACM workshop on Online social networks*, WOSN '09, pages 13–18, New York, NY, USA, 2009. ACM.

[50] C. Shapiro. Setting Compatibility Standards: Cooperation or Collusion? In R. Dreyfuss, D. Zimmerman, and H. First, editors, *Expanding the boundaries of intellectual property: innovation policy for the knowledge society*. Oxford University Press, 2001.

[51] T. Stein, E. Chen, and K. Mangla. Facebook immune system. In *Proceedings of the 4th Workshop on Social Network Systems*, SNS '11, pages 8:1–8:8, New York, NY, USA, 2011. ACM.

[52] P. Stuedi, I. Mohomed, M. Balakrishnan, Z. M. Mao, V. Ramasubramanian, D. Terry, and T. Wobber. Contrail: Enabling decentralized social networks on smartphones. In *Middleware*, pages 41–60, 2011.

[53] A. Tootoonchian, S. Saroiu, Y. Ganjali, and A. Wolman. Lockr: better privacy for social networks. In *CoNEXT*, pages 169–180, 2009.

[54] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas. Adnostic: Privacy preserving targeted advertising. In *In NDSS*, 2010.

[55] Vodafone. Rethinking Personal Data. http://www.vodafone.com/content/dam/vodafone/about/privacy/vodafone_rethinking_personaldata.pdf, 2011.

[56] W3C Federated Social Web Incubator Group. Social Web Acid Test - Level 0. http://www.w3.org/2005/Incubator/federatedsocialweb/wiki/SWAT0, May 2011.

[57] Wikipedia. CAP theorem. http://en.wikipedia.org/w/index.php?title=CAP_theorem&oldid=472196147. [Accessed 4-February-2012].

[58] Wikipedia. Distributed social network. [Accessed 4-February-2012]. http://en.wikipedia.org/w/index.php?title=Distributed_social_network&oldid=471838360.

[59] C. Wilson, T. Steinbauer, G. Wang, A. Sala, H. Zheng, and B. Y. Zhao. Privacy, availability and economics in the polaris mobile social network. In *Proc. of HotMobile*, Phoenix, AZ, March 2011.

[60] Wired. The Dawn of the Infomediary. http://www.wired.com/techbiz/media/news/1999/02/18094, 1999.

[61] J. Zittrain. *The Future of the Internet–And How to Stop It.* Yale University Press, New Haven, CT, USA, 2008.

Privacy Oriented Access Control for Electronic Health Records

Randike Gajanayake
Queensland University of Technology
Brisbane, Australia
g.gajanayake@qut.edu.au

Renato Iannella
NEHTA
Brisbane, Australia
renato.iannella@nehta.gov.au

Tony Sahama
Queensland University of Technology
Brisbane, Australia
t.sahama@qut.edu.au

ABSTRACT

Security and privacy in electronic health record systems have been hindering the growth of e-health systems since their emergence. The development of policies that satisfy the security and privacy requirements of different stakeholders in healthcare has proven to be difficult. But, these requirements have to be met if the systems developed are to succeed in achieving their intended goals. Access control is a fundamental security barrier for securing data in healthcare information systems. In this paper we present an access control model for electronic health records. We address patient privacy requirements, confidentiality of private information and the need for flexible access for health professionals for electronic health records. We carefully combine three existing access control models and present a novel access control model for EHRs which satisfies requirements of electronic health records.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection - *access control*

General Terms

Algorithms, Security

Keywords

Access control, MAC, DAC, RBAC, privacy, security, electronic health records, EHR

1. INTRODUCTION

Security of electronic health records (eHR) is a critical aspect of e-health solutions. Many different solutions have been developed over the years but the questions still remains as to whether the data in eHRs are secure enough. The National e-health transition authority (NEHTA) is the Australian authority dedicated to developing better ways of electronically collecting and securely exchanging health information. In their newest venture, the development of the personally controlled electronic health record (PCEHR) system, they have identified that privacy and security are major issues that need to be addressed properly in order for the proposed model to be well received [1]. Authentication is the initial stage of validation of the users to determine whether they are who they claim they are. Once authenticated, the users can enter an information system but access to information will still be governed by an access control policy. Access control is one of the main safeguards against improper data access. Access control aims to control the data usage of authorised users [2]. Access control models assume that the users are authorised to access the information system. After authorisation, the access control mechanism will define what information each authorised user can access. Many different access control models have been proposed and among them discretionary access control (DAC), mandatory access control (MAC) and role based access control (RBAC) are well established models.

Proper access control policies are a necessity for any EHR systems operation [1, 3]. Healthcare is an information dependant industry. The nature of the healthcare industry makes the access requirements different from other types of industries. Healthcare providers have data access requirements and the patients have data privacy requirements which may, in some instances, contradict the access requirements of the healthcare provider. Fulfilling all requirements is a complex task that has to be overcome in order to gain the confidence and trust of the end users of healthcare information systems.

In this paper we will introduce a privacy oriented access control model for electronic health records. The model is designed by combining the afore mentioned access control models with a purpose based access control (PBAC) mechanism for data access by authorised users. The purpose of the introduced access control model is to capture the different requirements of e-health into one module that can be adopted in a working electronic health records system.

2. RELATED WORK

In this section we will briefly introduce the access control models that have been considered in this paper. Even though these models have gone through many alterations and extensions, we will consider the basic principles behind each model so that is it easy to clarify how each model has been applied in our proposed access control model. Different access control strategies for e-health systems have also been developed in the past [3, 4]. Even though this work has been considered in developing the proposed model, due to space restrictions we will not discuss those techniques and approaches in this paper.

2.1 Discretionary access control

Discretionary Access Control uses access restriction set by the owner of the data object to restrict access to the objects. The

DUMW2012, April 16, 2012, Lyon, France.

users are bound by the authorizations which specify the operations each user can perform on specified objects such as read (R), write (W) and execute (EXE) [2]. The DAC model uses an access control matrix to assign access rights to users. A simple access control matrix is shown in Table 1.

Table 1. Access control matrix

User	Object 1	Object 2	Object 3	Object 4
Peter	R,W, EXE	R,W	-	R,W, EXE
Claudia	R,W	-	R,W, EXE	-
Bill	-	R,W, EXE	R, W	-
Matt	-	-	-	R,W, EXE

Implementing this matrix in large systems is a tedious task and representing it as a matrix will consume a considerable amount of resources. To represent this in a practical system the most common approach is by means of an Access Control List (ACL) and a Capability List (CL). An ACL is used to associate each object with the users who can access it. This association also contains the type of access (R, W, and EXE) to the object. This is a column wise representation of the access matrix. A Capability List is used to associate each user with the access permissions to the objects. This is a row wise representation of the access matrix.

DAC models have some inherent drawbacks. A significant issue is the fact that a user who is allowed to access an object by the owner of the object has the capability to pass on the access right to other users without the involvement of the owner of the object. This will create inevitable privacy issues if the DAC policy is used in an eHR system. Another factor we have to consider is the ownership of the data. In healthcare we cannot clearly identify a single entity as the owner of health data. An initial argument would be that the patients are the owners of their own health data. But patients are not always health professionals and it is likely that the involvement of a health authority of a relevant sort is necessary. Due to these reasons it is difficult to use only a DAC policy and fulfill access and privacy requirements of all healthcare stakeholders.

2.2 Mandatory access control

Mandatory access control systems do not consider the requirements of the owners of the data objects [5]. The access to data objects is controlled by assigning a security level to each object and comparing that security level to the user's security clearance and need-to-know. In order to access an object, the user must possess a clearance that is greater than or equal to the objects classification. In the MAC policy the flow of information from a higher security level to a lower security level is prevented by the "*Read Down*" and "*Write Up*" rules [2]. Similarly the integrity of the data objects can be protected by using the "*Read Up*" and "*Write Down*" Rules.

In a healthcare environment, we believe that assigning security levels to objects for the purpose of restricting access is not suitable. This is because the same type of data may have different sensitivity levels for different patients. We will discuss how we overcome this later in the paper.

2.3 Role based access control

Role base access control [6] models use permissions and rights that are assigned to roles in an organization to control access to data objects. It does not consider the access rights of an individual. Roles are assigned to all individual users in the systems. The users inherit the access permissions assigned to each role. This allows the system administrators to assign users to roles rather than go through the tedious task of assigning access rights to each and every user.

Roles are assigned to users depending on their capabilities and the job requirements within an organization. Each user must be given the least privilege depending on their job functions. RBAC policy uses the *need-to-know* principle to assign permissions to roles and to fulfill the least privilege condition.

2.4 Purpose based access control

According to the OECD guidelines, "*the purposes for which personal data are collected should be specified not later than at the time of data collection and the subsequent use limited to the fulfillment of those purposes or such others as are not incompatible with those purposes and as are specified on each occasion of change of purpose*" [7]. Purpose-based access control (PBAC) is bases on the notion of relating data objects with purposes [8]. These purposes can determine for what reason data is collected and what they can be used for. Much research has been done in this area and most have identified that greater privacy preservation is possible by assigning objects with purposes [8-10]. However, according to Al-Fedaghi [11], purpose management introduces a great deal of complexity at the access control level. Despite the complexity issues with PBAC, it can help capture the reasons for data collection as well as the intentions of the users, which is a vital factor in healthcare information systems where privacy preservation is a must.

3. ACCESS AND PRIVACY REQUIREMENTS OF EHE END USERS

Environments such as healthcare require security mechanisms that are different and more specialized than those applicable to other industries. Access control models that have been developed are insufficient to fulfill the requirements of eHR systems [12]. This is due to the convoluted nature of the industry and the nature of the information used. To address this issue a specialized access control model has to be designed taking in to consideration the different requirements of different users/entities involved.

In healthcare there are certain requirements that cannot be disregarded when developing an information system. In this section we will discuss those requirements with respect to healthcare providers and patients that have to be considered and addressed in terms of access control.

3.1 Access Requirements of Healthcare Providers

The following access requirements of healthcare providers (both individual and the health authority) can be identified that need to be addressed in the development of an information system.

1. A healthcare authority should have the capability to define their security policies within an organization.
2. Healthcare providers need easy access to the relevant information in a non restrictive and timely manner.
3. Healthcare providers need to have the capability to share patient health information with other health specialists to make well informed decisions.
4. A healthcare authority should have the power to override the patients' security settings in certain

circumstances. E.g. A life threatening emergency situation.

3.2 Privacy Requirements of a Patient

A patient's health information may contain sensitive information such as sexual health, mental health, addictions to drug or alcohol, abortions, etc. This makes such a patient demand strong security for their eHRs. These requirements however cannot contradict those set by the healthcare providers or the healthcare authority discussed above. If they do so the settings set by the health authority must prevail. A formal definition of this is given later in the paper. We note however, that in the PCEHR [1] system proposed by NEHTA, all privacy settings are set by the patients. Therefore such conflicts will not arise in their proposed system. The following capabilities can be identified as requirements of a patient with an eHR in terms of access control.

1. Patients need to have the capability to control access to their eHR. They should be able to allow only a preferred set of medical practitioners to access their eHR.
2. Patients need to be able to hide certain health information from health practitioners who already have access to their eHR.
3. Patients need to have the capability to see how their eHR is manipulated by users who have access to it.
4. The administration process of the security settings must be easy to understand and handle.

It is important to note that access restrictions might not always be beneficial to the patient. While fulfilling these privacy requirements under no circumstance must the patients' health be compromised.

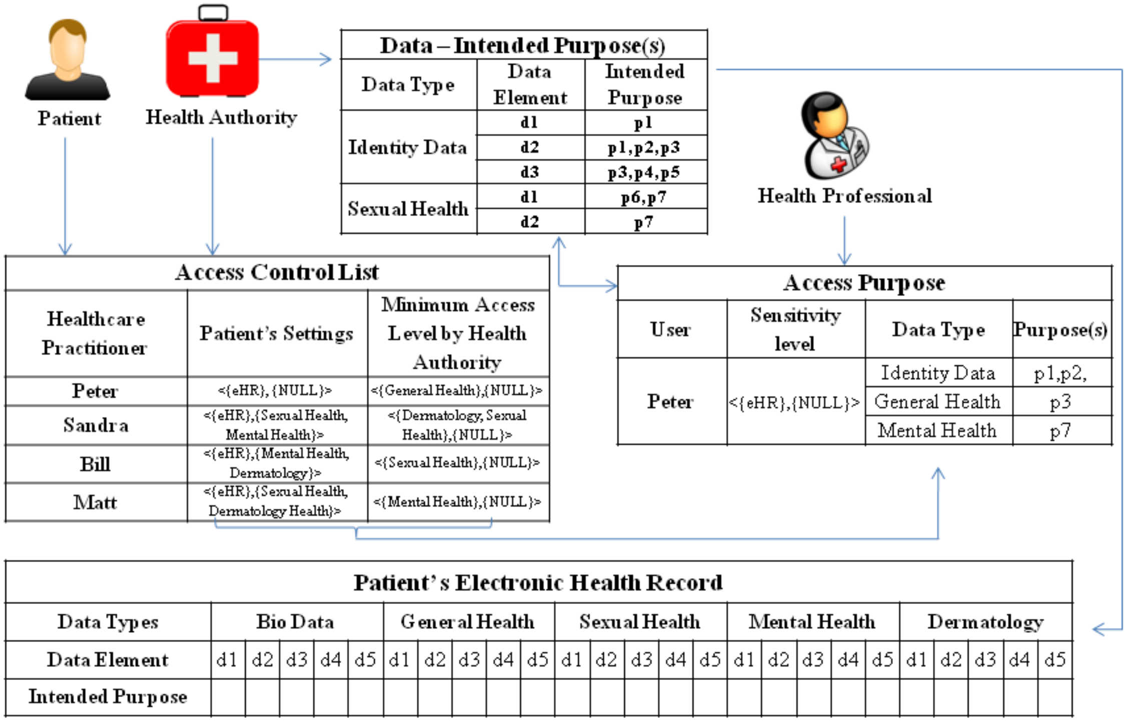
4. PROPOSED ACCESS CONTROL MODEL

The proposed model consists of four modules, a RBAC module, a MAC module, a DAC module and a PBAC module to fulfill the requirements of each of the stakeholders. The basic protocol for the proposed access control system is illustrated in Figure 1. We assume that the patient has a comprehensive eHR which is managed by a relevant health authority. In reality individuals may not want all information entered in to their eHR [1]. This requirement of course can easily be considered at the point of data entry. Nonetheless, we will show how a proper access control mechanism would eliminate the need to withhold information. In the proposed model the patient, the preferred healthcare providers and the health authority has certain operations and responsibilities to perform and fulfill.

Table 2. Data types and purposes

Data type	*Intended Purpose(s)*
Identity Data (PII)	p1
General Health	p1, p2, p3, p4
Sexual Health	p5
Mental Health	p5, p6, p7

The eHR is divided into data types (Table 2). Each data type in the eHR has to have a purpose or a set of related purposes. These are the intended purposes for which data is collected.



Data – Intended Purpose(s)

Data Type	Data Element	Intended Purpose
Identity Data	d1	p1
	d2	p1,p2,p3
	d3	p3,p4,p5
Sexual Health	d1	p6,p7
	d2	p7

Access Control List

Healthcare Practitioner	Patient's Settings	Minimum Access Level by Health Authority
Peter	<{eHR}, {NULL}>	<{General Health},{NULL}>
Sandra	<{eHR},{Sexual Health, Mental Health}>	<{Dermatology, Sexual Health},{NULL}>
Bill	<{eHR},{Mental Health, Dermatology}>	<{Sexual Health},{NULL}>
Matt	<{eHR},{Sexual Health, Dermatology Health}>	<{Mental Health},{NULL}>

Access Purpose

User	Sensitivity level	Data Type	Purpose(s)
Peter	<{eHR},{NULL}>	Identity Data	p1,p2,
		General Health	p3
		Mental Health	p7

Patient's Electronic Health Record

Data Types	Bio Data					General Health					Sexual Health					Mental Health					Dermatology				
Data Element	d1	d2	d3	d4	d5	d1	d2	d3	d4	d5	d1	d2	d3	d4	d5	d1	d2	d3	d4	d5	d1	d2	d3	d4	d5
Intended Purpose																									

Figure 1. Proposed access control architecture

Definition of purposes, without doubt, is a complicated task that requires much care. This process itself has to explore medical knowledge from medical professionals who can identify the significance of a single data element in the care giving process. Purpose definition itself has to be a system design phase since these purposes will govern the final access to data in the proposed access control model. The data types contain data elements related to them. In a more fine grained level purposes are related to data elements. For example, Identity Data of a patient can be divided into Name, Date of Birth, Age, Residential Address, etc. The Address can be further divided into street address, Town, State, Country and post code. The more detached the data field gets, the more fine grained it become. We will not go into details of how each data element is related to purposes in this paper. We shall leave that under future work and will simply assume that sufficient relationships exist between data elements and purposes.

The health authority will manage the relationship between data types and purposes. There will be a default set of purposes for every data type and elements of that data type. The health authority can define, add and remove purposes related to data types and elements. This will ensure that up to date purposes are maintained in the systems such that the access requirements of care providers are not wrongfully denied. It is understood that the proper definition of intended purposes is a key factor in this model. For the system to reach an optimum performance level it will undoubtedly take time in which initial purpose definitions would be altered and new purposes defined. The data elements in the eHR are also assigned a sensitivity label. This label will be used to determine who has clearance to access the data element. The overall description of the proposed protocol is given in the sections below using a case scenario.

4.1 Case scenario

Gary has a comprehensive eHR which is managed by a central healthcare authority.

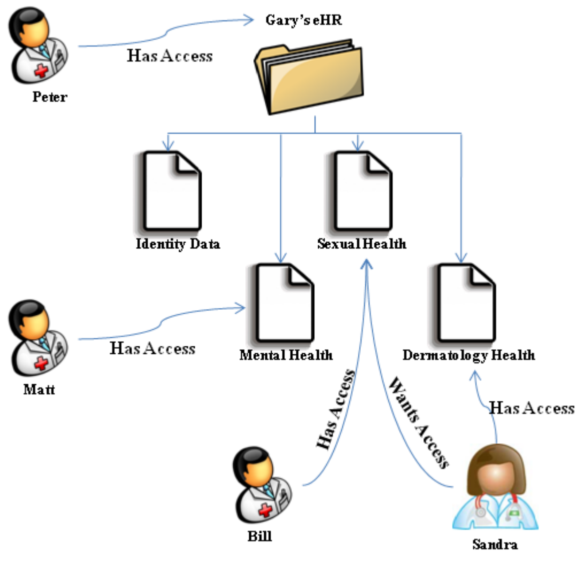


Figure 2. Case scenario

Gary's GP is Peter. As his GP, Gary has allowed Peter complete access to the data in his eHR. Gary has also been treated by Sandra a dermatologist, Bill a sexual health specialist and Matt a mental health specialist in the recent past. As a result Gary allows Bill to access his sexual health details, Matt to access his mental health details and Sandra to access his dermatology health details. He does not want Bill or Sandra accessing his mental health details and Matt or Sandra accessing his sexual health details. Gary suffers from a severe skin disease and does not want either Bill or Matt accessing his dermatology details due to embarrassment. He is aware that his care providers may need to share his information with other specialists but does not want them sharing the details without his consent. Sandra believes Gary's skin condition may be related to a known STD and wants access to Gary's sexual health details.

4.2 Role Based Access Control Module

In the RBAC module the healthcare authority will define the role structure of the health organization and assign the minimum access level for each role in the organization. In this role definition each role will be given a default sensitivity level for data access which will be discussed later. Even though the patients' privacy requirements have to be considered before data access is granted, there is no input from the patient for this module. The module is purely dedicated to fulfilling the organizational access and policy requirements. In a normal RBAC model, the role of the users has to change when the permissions for user changes. For this reason only the initial user-role assignment is done using the RBAC module.

4.3 Mandatory Access Control Module

In the MAC module, the health authority defines intended purposes for each data type and element. Deciding the sensitivity level of health information is a complex issue. The sensitivity labeling mentioned here are different from the classical hierarchical security levels found in MAC [2]. It is difficult to define a clear hierarchical structure for the sensitivity of data elements that is general to all participants. For example, sexual health and mental health information may have the same sensitivity label for some patients and may not be so for others. If a hierarchical structure is defined, it would be difficult to fulfill certain privacy requirements of patients.

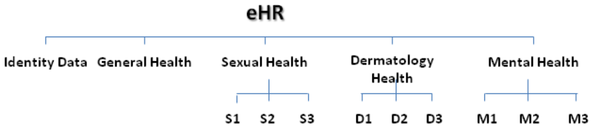


Figure 3. Object sensitivity tree

We propose sensitivity labeling of eHR data using a tree structure (Figure 3) that has the eHR itself as the root element, the data types as children and data elements as grandchildren. We use a similar technique introduced for purpose representation in Byun et al [13] to represent the sensitivity label of data elements in our model. We refer to this representation as the Sensitivity Tree (ST). A sensitivity label is not assigned to the objects themselves rather we relate the access level of a particular user in terms of the sensitivity label of the data elements.

Definition: A sensitivity label (SL) is a tuple <ASL, PSL>, where ASL = {asl_1, asl_2…asl_n} a set of allowed sensitivity labels and PSL = {psl_1, psl_2…psl_n} is a set of prohibited sensitivity labels.

ASL = {asl_i}; i = 1…n is denoted as all of the descendants of asl_i including asl_i.

PSL = {psl_j}; j = 1…n is denoted as all of the descendants of psl_j including psl_j.

Example: Matt can access to Gary's mental health details but cannot access his Sexual or Dermatology details. The access level for Matt can be represented in terms of sensitivity labels as follows.

SL_{Matt} = < {eHR}, {Sexual Health, Dermatology Health} >

Here we use the Denial-Takes-Precedence [14] principle. Access is granted to the entire eHR and then access is denied to specific field by the PSL. This helps isolate the most sensitive information in the eHR that need to be hidden from certain users. The access level for a particular user can also be represented as follows.

SL_{Matt} = < {Identity Data, General Health, Mental Health}, {NULL} >

Specifying the data elements that Matt can access can be a tedious task than specifying the data elements he cannot access. We will use this representation to represent the minimum access levels defined by the health authority. The health authority is only concerned with allowing access to particular data fields for the relevant health practitioners. This representation can also be used in purposes such as research where access is required only to a particular data type.

Example: Data of a survey of people who have suffered from some form of a STD during the last 10 years. For this purpose access is required only for the sexual health data type. Under no foreseeable circumstance would there be a requirement for accessing other fields of the eHR. The access level can be represented as follows.

$SL_{Researcher}$ = < {Sexual Health}, {NULL}>

Using this method of representing the access levels give enables more fine grained control over the data accessed by users.

4.4 Discretionary Access Control Module

In the DAC module the patient will specify who can access his eHR. He will populate an Access Control List (ACL) with the healthcare practitioners who he prefers to be able to access his eHR. The patient also has the capability to specify the access level of each of the users in terms of a sensitivity label in the ACL which is done using the MAC module as seen earlier.

Table 3. Access control list

Healthcare Practitioner	**Patient's Settings**	**Minimum Access Level Set by Health Authority**
Peter	<{eHR}, {NULL}>	<{General Health},{NULL}>
Sandra	<{eHR},{Sexual Health, Mental Health}>	<{Dermatology, Sexual Health},{NULL}>
Bill	<{eHR},{Mental Health, Dermatology}>	<{General Health, Sexual Health},{NULL}>
Matt	<{eHR},{Sexual Health, Dermatology Health}>	<{General Health, Mental Health},{NULL}>

The table above shows an abstract ACL. Gary has granted 4 health care practitioners access to his eHR. But the access is bound by the patient's privacy settings and the settings by the health authority. The settings by the health authority are set during the role assignment in the RBAC module.

The sensitivity level defined by the health authority is different to what is defined by the patients. PSLs set by the health authority will always be *NULL*. As mentioned above, this is because the health authority is concerned with allowing access to the health professionals. The prohibitions are defined by the patients. The allowed sensitivity level set by the patients always precedes that which is set by the health authority if there is no conflict between the patients prohibited sensitivity label and the allowed sensitivity label set by the health authority. The allowed sensitivity level set by the health authority always precedes the prohibited sensitivity label set by the patients if there is a conflict. This characteristic/notion will ensure that the relevant information is always available to the right person in terms of providing better healthcare. A formal definition for this notion is given below.

Definition:

- IF ($ASL_{Patient} \geq ASL_{HealthAuthority}$ AND $PSL_{Patient} \cap ASL_{HealthAuthority} = \emptyset$) THEN $SL_{HealthProfessional}$ = < {$ASL_{Patient}$, {$PSL_{Patient}$} >
- IF ($ASL_{Patient} \leq ASL_{HealthAuthority}$ AND $PSL_{Patient} \cap ASL_{HealthAuthority} = \emptyset$) THEN $SL_{HealthProfessional}$ = < {$ASL_{HealthAuthority}$}, {$PSL_{Patient}$} >
- IF ($ASL_{Patient} \geq ASL_{HealthAuthority}$ AND $PSL_{Patient} \cap ASL_{HealthAuthority} \neq \emptyset$) THEN $SL_{HealthProfessional}$ = < {$ASL_{Patient}$}, {$PSL_{Patient} \cap ASL_{HealthAuthority}$`} >

When these conditions are satisfied, the sensitivity levels are updated so that the users can access the relevant data types/elements. E.g. Sandra (Table 4) will be assigned a sensitivity level SL_{Sandra} = < {eHR}, {Mental Health}>.

Algorithm 1 shows how sensitivity levels are set for the users. The symbols other than the ones used previously denote as follows. P_{SL} and HA_{SL} denote sensitivity levels set by the Patient (*P*) and the Health Authority (*HA*) respectively.

Algorithm 1: Set Sensitivity Label SL_{UID}

```
Input: 1. User ID: $UID$
       2. Access Control List: $ACL$
Output: User Sensitivity Label $SL_{UID}$
Method:
    $P_{SL\_UID} \leftarrow$ <$ASL_{P\_UID}$, $PSL_{P\_UID}$>
    $HA_{SL\_UID} \leftarrow$ <$ASL_{HA\_UID}$, $PSL_{HA\_UID}$>
    if ($ASL_{P\_UID} \geq ASL_{HA\_UID}$ AND $PSL_{P\_UID} \cap ASL_{HA\_UID} = \emptyset$) then
        $SL_{UID} \leftarrow$ < {$ASL_{P\_UID}$}, {$PSL_{P\_UID}$} >
        else if ($ASL_{P\_UID} \leq ASL_{HA\_UID}$ AND $PSL_{P\_UID} \cap ASL_{HA\_UID} = \emptyset$) then
            $SL_{UID} \leftarrow$ < {$ASL_{HA\_UID}$}, {$PSL_{P\_UID}$} >
        else if ($ASL_{P\_UID} \geq ASL_{HA\_UID}$ AND $PSL_{P\_UID} \cap ASL_{HA\_UID} \neq \emptyset$) then
            $SL_{UID} \leftarrow$ < {$ASL_{P\_UID}$}, {$PSL_{P\_UID} \cap ASL_{HA\_UID}$ `} >
        else if ($ASL_{P\_UID} \leq ASL_{HA\_UID}$ AND $PSL_{P\_UID} \cap ASL_{HA\_UID} \neq \emptyset$) then
            $SL_{UID} \leftarrow$ < {$ASL_{HA\_UID}$}, {$PSL_{P\_UID} \cap ASL_{HA\_UID}$ `} >
    end if
    return $SL_{UID}$
```

4.5 Purpose Based Access Control Module

This module primarily deals with the access requests of authorised users. When a user requires access to data in an eHR they define an access request consisting the reason(s) or purpose(s). This definition will be compared to the purposes in Table 2 which were assigned to the data elements by the health authority and if satisfied access will be granted.

Table 4 represents typical access requests by authorised health practitioners. An access request may not particularly be for a single task. And each data type requested may not always be associated with a single purpose. The users must have the capability to specify multiple purposes in a single access request to enhance the ease of use. If access is granted we have to make the assumption that each data element can only be used for the specified access purpose(s). The health information systems which would use this access control model should have the capability to provide the functionality where data misuse can be captured.

Algorithm 2: Access Request

```
Input:  1. User ID: UID
        2. Sensitivity Level: $SL_{UID}$
        3. Access Purposes List: $AccPurList[d_{AP}, p_{AP}]$
        4. Access Control List: ACL
        5. Intended Purposes List: $IntPurList\ [d_{IP}, p_{IP}]$
Output: Access_State []
Method:
 Num_Requests ← Size (AccPurList)
 Access_State [Num_Requests] ← False
 Permit_Data [Num_Requests] ← False
 Check_Purpose [Num_Requests, Num_Pur] ← False
    for i = 1 to Num_Requests do
        if $IntPurList(i) \in PSL(SL_{UID})$ then
            Permit_Data[i] ← False
        else
            Permit_Data[i] ← True
        end if
    for j = 1 to Size(AccPurList(i)) do
        if $AccPurList[i, j] \subseteq IntPurList$ then
            Check_Purpose [i, j] ← True
        else
            Check_Purpose [i, j] ← False
        if {(Permit_Data [i] = True) AND
    (Check_Purpose [i, j] = True ) = True} then
            Access_State [i] ← True
        else
            Access_State [i] ← False
        end if
    end for
    end for
    return Access_State []
```

Algorithm 2 processes the access requests by health professionals. A tuple with data type and purpose is denoted as *<d, p>*. *Permit_Data []* contains the status (allowed or disallowed) of the data types requested by the user. *Check_Purpose [Num_Requests, Num_Pur]* is a 2D array containing the status of the purposes for each the data type requested. The algorithm returns and array *Access_State []* with the state of each purpose in the access request. *IntPurList* $[d_{IP}, p_{IP}]$ is a 2D array with data types with their intended purposes (set by the health authority). *AccPurList* $[d_{IP}, p_{IP}]$ is a 2D array with the data types and their access purposes (requested by a user)

Table 4. Access requests by authorised users

User	*Sensitivity level*	*Data Type (d)*	*Access Purpose (p)*
Peter	<{eHR},{NULL}>	Identity Data	p1,p2
		General Health	p3
		Mental Health	p7, p4
		Sexual Health	p5
Sandra	<{eHR},{Mental Health}>	Dermatology	p8
		Sexual Health	p5

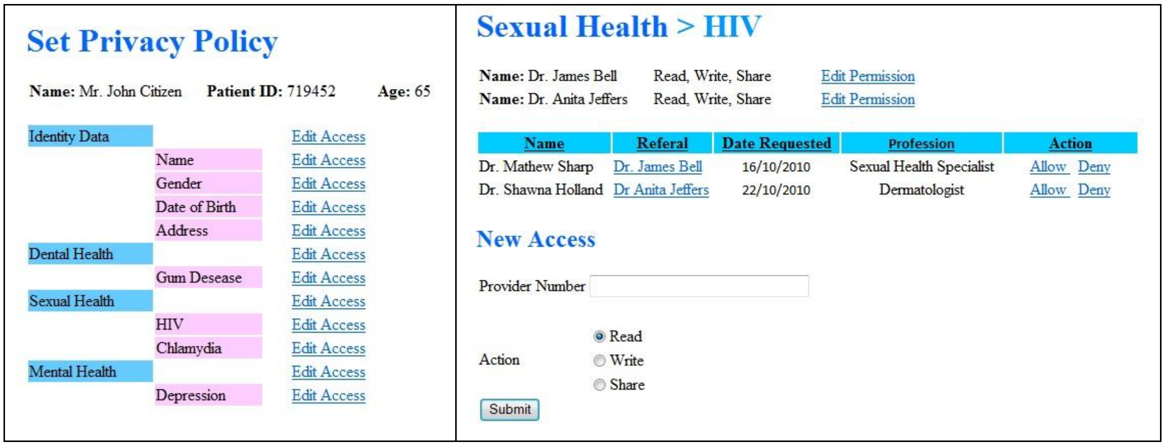
It is important to note that the nature of the healthcare industry force us to adopt the *break the glass* emergency mechanisms where the patients health prevails over privacy requirements. Also, usability is a vital part of every healthcare information system. No matter what the underlying principles are, the users, both patients and the healthcare providers must be given simple directions (e.g. menu) where they can set their access settings easily.

4.6 Information Sharing Example

In our case scenario let us assume that Peter, using the PBAC module defined within the portal for authorised users, initiates a request to share Gary's sexual health details with another health professional Claudia for the benefit of Gary. Here however, Claudia should have the relevant access clearance by the health authority to access the type of data specified by the requester. This default access level is set using the RBAC and MAC modules of the access control model. It is not necessarily required that the receiving health professional be in Gary's ACL which is defined by Gary through the DAC and MAC modules since it is a request by an authenticated user. It is important to note that Gary's consent for sharing information is already given to Peter by the policies set by the patient and the health authority. Gary can give any health professional the right to share his health information without his consent with other health professionals. If Claudia accepts the request she becomes an authorised user of the system with the relevant access level. Gary has the right to remove Claudia from the ACL at a later time. Gary is notified of the actions of the users at relevant times to make the system transparent. It is important to note that information is shared for the benefit of the patient. Information must not be misused by the users. Trust plays a major role in the information sharing process. Furthermore, such processes are traceable and accountable. An eHR system using this protocol must have the capability to prevent users from misusing information.

5. PROTOTYPE

A prototype of the proposed access control model was developed. The prototype is a Web based system aimed at testing the proposed protocol. A Web based prototype was developed because with extensions, information accountability systems with reasoning capabilities such as the one proposed by Gajanayake et. al [15] can be developed.



Set Privacy Policy

Name: Mr. John Citizen **Patient ID:** 719452 **Age:** 65

Identity Data		Edit Access
	Name	Edit Access
	Gender	Edit Access
	Date of Birth	Edit Access
	Address	Edit Access
Dental Health		Edit Access
	Gum Desease	Edit Access
Sexual Health		Edit Access
	HIV	Edit Access
	Chlamydia	Edit Access
Mental Health		Edit Access
	Depression	Edit Access

Sexual Health > HIV

Name: Dr. James Bell Read, Write, Share Edit Permission
Name: Dr. Anita Jeffers Read, Write, Share Edit Permission

Name	Referal	Date Requested	Profession	Action
Dr. Mathew Sharp	Dr. James Bell	16/10/2010	Sexual Health Specialist	Allow Deny
Dr. Shawna Holland	Dr Anita Jeffers	22/10/2010	Dermatologist	Allow Deny

New Access

Provider Number

Action ● Read ○ Write ○ Share

Figure 3. Left: patients can allow or deny access to data types for health professionals Right: patients can view current health professionals who has access to particular data elements and can assign new health professionals to access the data elements

This implementation is focused only on demonstrating the proposed access control protocol. We are not focused on actual system usability at this stage. Figure 3 shows a portion of the prototype that allows patients to set and manage their privacy policies.

The prototype is developed to handle three types of users; patients, health authority and health professionals. The patients and the health authority can set privacy and access policies and the final policies are formulated according to the protocol discussed above. A simple SQL database is used to hold the policies and the data in the eHR. Health professionals can lodge access requests which consist of access purposes and will be processed according to the protocol using an intended purposes database managed by the health authority. The management of intended purposes is not facilitated in this prototype.

6. DISCUSSION

Access control has been a fundamental security measure of information systems for many years. Amongst many different models DAC, MAC, RBAC and PBAC are the most popular. These models come in many different variations and are used in different contextual domains. In this paper we discussed how we can make use of the characteristics and principles of these models to facilitate a suitable access control model for electronic health records. We identified specific requirements of different healthcare stakeholders and combined the principles behind the DAC, MAC, RBAC and PBAC models to address them. The DAC model is used to capture the access settings for users by patients. Patients maintain an ACL of their trusted health professionals and use a variation of the MAC model to assign access levels (or sensitivity level as discussed above) for them. The MAC model is used to define access levels of health professionals who can access data in an eHR. A central health authority uses a RBAC model and the MAC model to set default access levels for health professionals. A simple PBAC model is used as a usage control module to capture the access purposes of information users. The current prototype is capable of demonstrating the process of setting the access levels by the patients and the health authority and processing access requests by health professionals. We have tested the prototype to demonstrate various scenarios of policy settings and data access. Further development and testing is required to investigate how this model would behave in the complex domain of healthcare. Rather than being used as a standalone security model, the final goal of our research is to harmonize the access control model with an information accountability framework (IAF) for e-health. The IAF uses DRM technologies to represent the access and usage policies set by the users in a Rights Expression Language [16].

7. CONCLUSION AND FUTURE WORK

In this paper we have introduced a novel access control model for electronic health record systems using prominent access control models. We have identified certain requirements of end users of an electronic health record system and our proposed model is designed to fulfill those requirements. Further to what has been discussed in this paper we propose the following additions. Purpose definition is an important part in our model. Building a comprehensive set of purposes and maintaining them is vital. These definitions must capture medical knowledge as well as system requirements. The health details of family members and relatives are an important resource that must be available to the caring professional. We intend to extend the proposed model such that those links can also be incorporated in to the model while still maintaining the integrity of the privacy capability of the model. We are also working to extend the proposed model to support explicit actions as described in [17] and providing non-restrictive access to health information for the authorized persons while incorporating an information accountability framework [15] so that health information would not be misused. Proper representation of policies is vital for such systems. We have extended the proposed access control model such that the policies are represented in a suitable rights expression language, namely the open digital rights language (ODRL) [18]. In this extended work we introduce an information accountability framework with policy reasoning capabilities which adheres to information accountability principles.

8. ACKNOWLEDGEMENTS

We would like to thank the National Information and Communications Technology Australia (NICTA) for partially funding this ongoing research project.

9. REFERENCES

[1] National E-Health Transition Authority. *Draft Concept of Operations: Relating to the introduction of a personally controlled electronic health record (PCEHR) system*. 2011.

[2] Sandhu, R. S. and Samarati, P. Access control: principle and practice. *Communications Magazine, IEEE*, 32, 9 1994), 40-48.

[3] Motta, G. H. M. B. and Furuie, S. S. A contextual role-based access control authorization model for electronic patient record. *Information Technology in Biomedicine, IEEE Transactions on*, 7, 3 2003), 202-207.

[4] Alhaqbani, B. and Fidge, C. Access control requirements for processing electronic health records. In *Proceedings of the Proceedings of the 2007 international conference on Business process management* (Brisbane, Australia, 2008). Springer-Verlag.

[5] Ferraiolo, D., Kuhn, D. R. and Chandramouli, R. *Role-based access control*. Artech House, 2003.

[6] Sandhu, R. S., Coyne, E. J., Feinstein, H. L. and Youman, C. E. Role-based access control models. *Computer*, 29, 2 1996), 38-47.

[7] OECD *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*. City, 1980.

[8] Byun, J.-W., Bertino, E. and Li, N. Purpose based access control of complex data for privacy protection. In *Proceedings of the Proceedings of the tenth ACM symposium on Access control models and technologies* (Stockholm, Sweden, 2005).

[9] Naikuo, Y., Howard, B. and Ning, Z. *A Purpose-Based Access Control Model*. City, 2007.

[10] Ni, Q., Bertino, E., Lobo, J., Brodie, C., Karat, C.-M., Karat, J. and Trombetta, A. Privacy-aware role-based access control. *ACM Trans. Inf. Syst. Secur.*, 13, 3 2010), 1-31.

[11] Al-Fedaghi, S. S. Beyond purpose-based privacy access control. In *Proceedings of the Proceedings of the eighteenth conference on Australasian database - Volume 63* (Ballarat, Victoria, Australia, 2007). Australian Computer Society, Inc.

[12] Finance, B., Medjdoub, S. and Pucheral, P. *Privacy of medical records: from law principles to practice*. City, 2005.

[13] Byun, J.-W. and Li, N. Purpose based access control for privacy protection in relational database systems. *The VLDB Journal*, 17, 4 2008), 603-619.

[14] Bertino, E. Data security. *Data & Knowledge Engineering*, 25, 1-2 1998), 199-216.

[15] Gajanayake, R., Iannella, R. and Sahama, T. Sharing with Care: An Information Accountability Perspective. *Internet Computing, IEEE*, 15, 4 2011), 31-38.

[16] Gajanayake, R., Iannella, R. and Sahama, T. An Information Accountability Framework for Shared E-Health Policies. In *Proceedings of the Workshop on Data Usage Management on the Web* (Lyon, France, 2012).

[17] HL7 *Role Based Access Control (RBAC) Role Engineering Overview*. City, 2010.

[18] ODRL Initiative *ODRL V2.0 - Core Model - Working Draft*. City, 2012.

Behavioural Control

David W Chadwick
University of Kent
School of Computing
Canterbury, UK
+44 122782 3221
d.w.chadwick@kent.ac.uk

Christopher Bailey
University of Kent
School of Computing
Canterbury, UK
+44 122782 3628
c.bailey@kent.ac.uk

Rogério de Lemos
University of Kent
School of Computing
Canterbury, UK
+44 122782 3628
r.delemos@kent.ac.uk

ABSTRACT

We describe the self-adaptive authorization framework (SAAF), an autonomic self-adapting system for federated RBAC/ABAC authorization infrastructures. SAAF monitors the behaviour of users, and when it detects abnormal behaviour, it responds by adapting the authorization infrastructure to prevent any further abnormal behaviour. The models and components of SAAF are described, as well as the current limitations and where future research is still needed.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: *Access controls*;

General Terms

Management, Security.

Keywords

Self-adaptation, authorization, autonomic access control, computing security, RBAC, ABAC, behavioural control.

1. INTRODUCTION

Usage control seeks to control the use of a particular resource after its initial access, so that future accesses are also controlled [1]. In this respect it is similar to digital rights management [2]. In this position paper we take a broader look at controlling the use of resources, through analysing users' behaviour. By monitoring all accesses to all resources, we can determine when a series of accesses, by one or more users, becomes outside the expected norms of behaviour. Our system then stops this abnormal behaviour by automatically adapting the access control system so that further abnormal or abusive behaviour is prevented. Our system is thus an example of an autonomic access control system, which is self-monitoring, self-adapting, and self-correcting.

1.1 Motivation

Our work is in part motivated by the case of Private Bradley Manning. During July 2010 it is alleged that Private Manning, a US army intelligence analyst, downloaded over 0.25 million classified US military documents from a US Department of Defence website [3]. Assuming that the US intelligence analyst was an authorized user and that access was requested and granted on a document-by-document basis, we can say that the analyst had appropriate access rights and that the authorization system performed its function correctly. Any monitoring of the authorization system on a request by request basis would not pick up any abnormal behaviour as it processed the analyst's access requests according to its access control policies. Usage control would similarly not have detected any usage problems on any single file, assuming an analyst was allowed to copy an accessed file onto a memory stick for further study and later analysis. Even if usage control had detected a usage problem, such as copying to a memory stick, and had forbidden it, no further action would have been taken even after the multiple occurrences of such events. However to a human administrator, monitoring the system use in real time, numerous similar requests from the same user to access different files in a short period of time would have flagged up inappropriate behaviour.

Unfortunately the cost of performing real time human monitoring is prohibitively expensive in most cases. Furthermore, making rapid changes to the system to stop further abuses is much more problematical for a human administrator. Analysing the misbehaviour, determining the course of corrective action to take, and then activating the chosen actions, might have taken a human administrator a significant amount of time, compared to the speed that a computer can do this. Consequently our research proposes to build an autonomic self-adaptive access control system that can automatically detect abnormal access control behaviour and apply corrective actions to the authorisation infrastructure. We call our system SAAF – A Self-Adaptive Authorization Framework – for policy based authorization systems. Note that SAAF is designed to be able to both restrict and enable user access, when abnormal behaviour is detected. An example of enabling user access, would be when a doctor has break the glass access rights to any patients' records, and indicates on breaking the glass for a patient's record that he now has a therapeutic relationship with that patient. SAAF would update the patient's record to record the new relationship, so that break the glass would no longer be needed.

This paper is an update of our SAAF, which was originally described here [4].

The rest of this position paper is structured as follows. Section 2 describes our models: that of the underlying federated RBAC/ABAC authorisation system, and that of the self-adaptive authorization framework that manages this. Section 3 concludes with a discussion of the current limitations and the research that

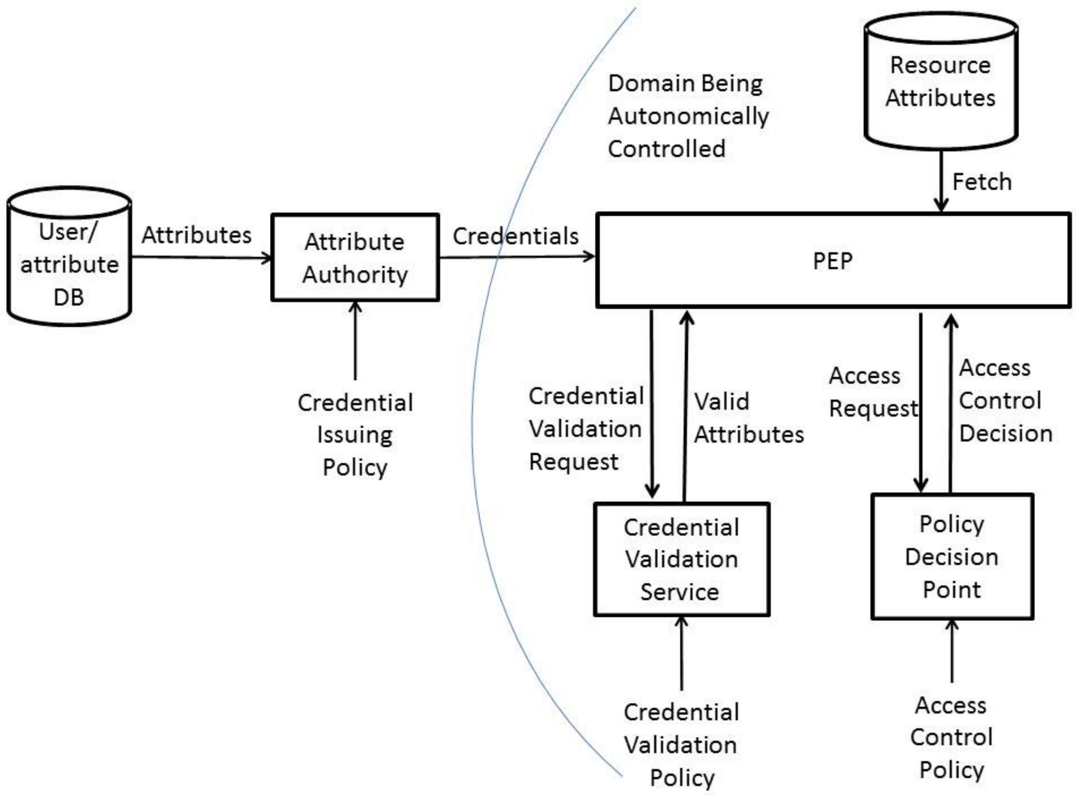


Figure 1. The Federated RBAC/ABAC Model.

still needs to be done in order to build a fully functioning prototype SAAF.

2. MODELS

2.1 Federated RBAC/ABAC Model

Figure 1 shows our model of a federated RBAC/ABAC system that we wish to autonomically control. In this model, we only show the objects that are relevant to of our autonomic access control system. We do not show users, since the system does not actually directly control them. Instead, it controls the *access* of users to resources, via the following system components:

- the Attribute Authorities that assign role/attribute credentials to users,
- the Credential Validation Service that validates user credentials,
- the resource attributes (metadata) that hold user information, and
- the Policy Decision Point which grants or denies users access to resources.

In a federated system, attribute authorities (AA) in different domains hold sets of user attributes in their locally managed databases. When a user wishes to access a federated resource from his web browser, the resource owner or service provider (SP) typically redirects the user's browser to the AA, which authenticates the user then assigns the user a digitally signed attribute assertion (or credential) according to its local Credential Issuing Policy. In Shibboleth [5], for example, this policy comprises the attribute release policies of both the user and the AA.

In a federated system that is capable of attribute aggregation the user may obtain several credentials from different AAs before attempting to gain access to the SP's resources, or the SP may pull credentials from various AAs during the process of granting access. Note that figure 1 does not show the actual protocol messages or web message flows, but only the logical flow of objects that are to be controlled by the autonomic system.

The user's browser presents his/her credentials to the SP's Policy Enforcement Point (PEP) in order to gain access to the SP's resources. The PEP validates these credentials by passing a credential validation request to its locally trusted Credential Validation Service (CVS), and receiving a set of valid attributes in return. The CVS is controlled by the SP's Credential Validation Policy that provides the rules for determining which AAs are trusted to assign which attributes to which users. This is the process of validating the user-role assignments from the traditional RBAC model.

The PEP fetches the attributes of the requested resource, and passes these, along with the user's valid attributes, to the Policy Decision Point (PDP) via an access request. The PDP grants or denies the user access to the requested resources according to its access control policy. This is the process of validating the role-permission assignments from the traditional RBAC model. The PDP returns its access control decision to the PEP, which then acts accordingly.

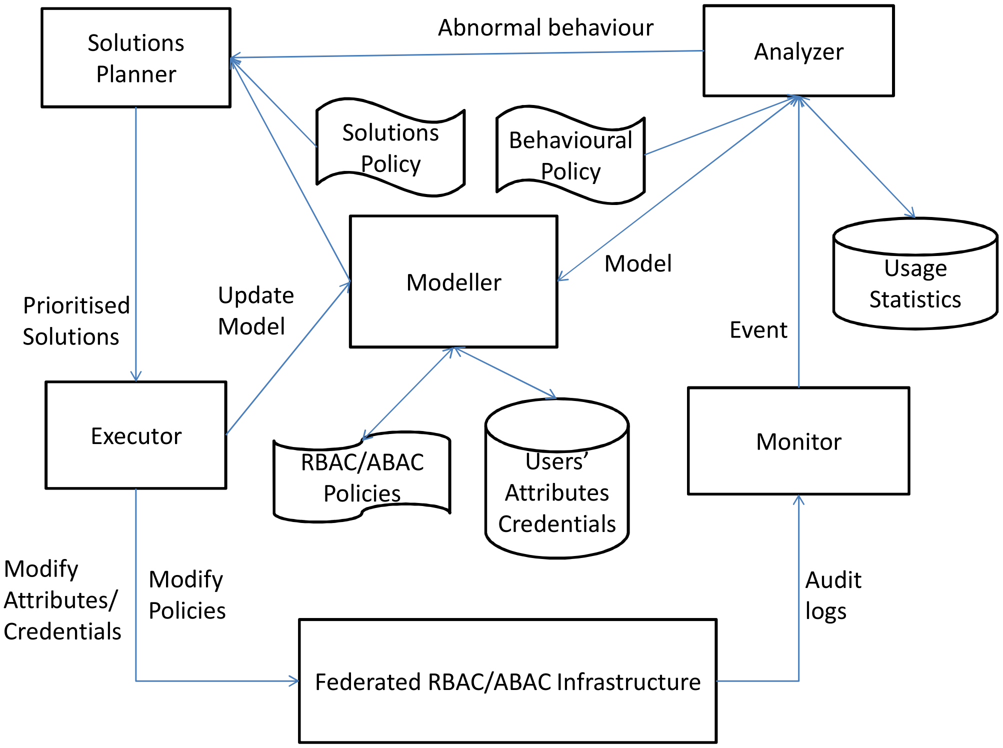


Figure 2. SAAF Components

The three policies, resource attributes, user attributes and user credentials of the RBAC/ABAC system are the six assets that our self-adaptive access control framework (SAAF) will automatically control.

We assume that the SP/PEP records in some locally secure audit log both its requests to the CVS and PDP and their responses. These log records will be used by SAAF to monitor the behaviour of the federated RBAC/ABAC system.

2.2 Self Adaptive Authorisation Framework (SAAF)

Figure 2 shows the components of our proposed self-adaptive authorisation framework. A federated RBAC/ABAC infrastructure that conforms to the federated RBAC/ABAC model presented above, becomes a single component of SAAF. It produces audit logs and is controlled by its policies as described above. SAAF monitors the behaviour of the users of the federated RBAC/ABAC system by inspecting its logs. When SAAF detects abnormal user behaviour it will attempt to alter this by enacting one or more solutions, which will modify the assets of the RBAC/ABAC system, as described below.

The Modeller contains a model of the assets of the actual RBAC/ABAC system that is being autonomically controlled, modelling the 6 assets shown in Figure 1. If the actual system does not have an asset from the federated RBAC/ABAC model, e.g. no credential validation policy, then this will be reflected in SAAF's asset model. Whilst different

RBAC/ABAC systems will use different policy languages to construct their policies e.g. XACML [8], PERMIS [9], the Modeller uses an abstract representation of these, using model transformations based upon an OWL ontology that we developed in a previous project when writing natural language access control policies (which are themselves policy language independent) [7]. Each of the policies needs to be reproduced in SAAF's model, so that SAAF's asset model reflects the actual RBAC/ABAC system being controlled. As changes are made to the underlying policies, SAAF's view of the RBAC/ABAC policies is kept synchronised with them (by the Executor). We do not expect to duplicate each of the AA's user/attribute databases in SAAF's model. Instead SAAF will be initialised with as much information as each AA is willing to release, which in the worst case could be nothing. As each user tries to access one of the SP's resources, the Monitor will detect this from the audit logs and notify the Analyser. If the Analyser notices that this user/role/attribute/credential is not in SAAF' user database it can add it, so that the user database will grow with time to reflect the AAs' databases. Similarly SAAF can be provided with a model of the SP's resource attribute database, or it can build it itself from the audit logs.

The Monitor component of SAAF is responsible for monitoring the usage of the RBAC/ABAC infrastructure, by reading in the audit logs, in their proprietary format, and extracting from them the events which are of interest to the SAAF Analyser, such as role X accessed resource Y at time t. Depending on the infrastructure of the federated environment, SAAF may use multiple Monitors to gain the information it needs. Each Monitor will be specific to its target application. These events are passed to the SAAF Analyser.

The Analyser keeps a usage statistics database that records the frequency of the various events that are passed to it. One event may produce several sets of statistics, such as the number of accesses a particular role or user has performed in the last minute/hour/day, the frequency a resource has been accessed, the total number of grants per time period etc. The Analyser is controlled by a Behavioural Policy set by the SP administrator (see Figure 3). This provides the behavioural norms of the RBAC/ABAC system, such as: the number of accesses by a role per minute/hour/day, the frequency of access to a particular resource, the number of invalid credentials that are received per time period, the frequency of system grants and denies, etc. Each behavioural norm has an associated cost, which represents the cost to the SP of the norm being exceeded, and of no corrective action being taken. The Analyser determines if the users of the RBAC/ABAC system are behaving as expected or not, as determined by the behavioural policy. This is akin to behavioural analysis performed in intrusion detection systems (IDSs) [6]. Abnormal behaviour could be due to several different reasons, such as wrongly specified policies in the RBAC/ABAC system, misuse of resources by authorised people, or attacks by unauthorised people. If the Analyser determines that abnormal behaviour has occurred it informs the Solutions Planner about this (see Figure 4).

The Solutions Planner is driven by a solutions policy, set by the SP administrator, which contains the various solutions that are available to counteract the detected abnormal behaviour. For example, abusive user behaviour can be counteracted by denying the abusive user(s) further access to the SP's resource. Federated users can be denied access to a federated resource via any of the following actions:

- removing the user's attributes from the AA's database
- modifying the AA's credential issuing policy
- revoking a user's already issued credentials
- removing resource attributes which identify the user
- modifying the SP's credential validation policy
- modifying the SP's access control policy

Each of these solutions has an associated cost. For example, revoking the credentials of a single user is far less costly to the SP than modifying the PDP's access control policy so as to deny all users access to the abused resource(s). The SP administrator is required to place a cost against each of the proposed solutions, so that they can be compared to the cost of the detected abnormal behaviour. The Solutions Planner needs to compare the Solutions Policy against the model of the federated RBAC/ABAC system, as held by the modeller, in order to draw up a list of prioritised solutions which are more cost effective than leaving the system alone. It may be that in some cases of minor abnormal behaviour, such as a student downloading dozens of journal papers in a few minutes, the cost of preventing the abnormal behaviour is greater than the cost of the abuse, and so no corrective action will be taken. However, if the abuse were to continue in a sustained fashion, then at some point it would become cost effective to take the corrective action, for example, once the student's downloads exceed a hundred journal papers per hour. The Solutions Planner sends its prioritised list of cost effective solutions to the Executor.

The role of the Executor is to implement the most cost effective solution, but if this fails, to implement the next highest priority one until one succeeds. The Executor comprises an Orchestrator and many different Interface Components (ICs) that communicate with their respective components of the RBAC/ABAC infrastructure. The Orchestrator converts the most cost effective solution into a set of instructions, which it sends to the ICs that are capable of modifying the various components of the federated authorization infrastructure. Once a solution has been completed and executed by all relevant ICs the Orchestrator updates SAAF's asset model to ensure that SAAF has a synchronised view of the actual RBAC/ABAC authorization infrastructure. The Executor needs to know the specific protocols, policy languages etc. being used by the monitored RBAC/ABAC system so that it can incorporate the correct ICs.

Some of the RBAC/ABAC assets being controlled are held in the SP's local domain, and therefore SAAF can be given permission to modify these directly. However, some of the assets belong to the domains of the remote AAs (i.e. the credential issuing policies and users' attributes and credentials), and therefore SAAF would not normally have permission to modify these. We propose to solve this in the following way. As part of the federation agreement, an AA must either give the SP's SAAF permission to directly update its assets (i.e. credential issuing policy, user attribute database or credential revocation list) or agree to provide a web listening service for SAAF to send update messages to, and to respond to these updates with confirmation messages within a specified time period. In this way the Executor can either directly perform the solutions itself, or can notify the remote AA of the required solution, then wait for the specified time for a response. If no response is received in the specified time it can determine that the solution has failed to be enacted and can move to the next solution in the list.

Note that only the Executor and the Monitor are dependent upon the implementation details of the monitored RBAC/ABAC system, as all the other SAAF components use their own internal formats for modelling the RBAC/ABAC system, recording usage statistics and specifying their policies. Thus the majority of SAAF is independent of the implementation details of the RBAC/ABAC system that is being controlled, allowing SAAF to be usable with different implementations of RBAC/ABAC through the implementation of application specific Monitors and Executor ICs.

3. DISCUSSION, LIMITATIONS, CONCLUSION

This position paper presents our current research on "behavioural control", which is an attempt to monitor and autonomically control the behaviour of users within a federated RBAC/ABAC authorisation system. The research is still at an early stage. To date we have concentrated on specifying the models, their essential components, and the authorisation assets that can be managed in order to control users' behaviour.

Modelling work that is still required is to:

- Specify the Behavioural Policy in detail,
- Specify the Abnormal Behaviour in detail,
- Specify the Solutions Policy in detail,
- Determine the full set of statistics that need to be recorded
- Specify the algorithms for determining abnormal behaviour and determining solutions.

We have to determine which semantics and rules the behavioural policy language will support based on the complexity of the constructs and the time it will take to evaluate the rules against the monitored behaviour.

```
<BhrRule id="FreqGetSameRes">
 <Resource>"+"</Resource>
 <Action>Get</Action>
 <Op>GT</Op>
 <Rate>
   <Number>5</Number>
   <Time>1</Time>
   <Unit>Min</Unit>
   <Cost>250</Cost>
 </Rate>
 <Rate>
   <Number>20</Number>
   <Time>1</Time>
   <Unit>Day</Unit>
   <Cost>500</Cost>
 </Rate>
</BhrRule>
```

Figure 3. An Example Behavioural Rule

An example behavioural rule is given in Figure 3. This states that the rate of requests for the Get action on the same resource (indicated by "+") must be no greater than 5 requests per minute, or 20 requests per day, and the cost of violating each rule is 250 and 500 units respectively. This is a very simple behavioural rule. More complex rules may involve specifying sequences of actions, such as downloading a file followed by copying it to a flash disk, on the same or different resources. Even more complex rules may involve identifying the same sets of actions being carried out by different users. Significant research is still needed in this area.

Figure 4 shows an example of flagging abusive abnormal behaviour. This signals which subjects (identified by their attributes) have performed which abnormal actions on which resources, and what the cost of this is to the organisation. In this example one user, a student with ID 123456, from Kent, has performed abusive Get actions on two different resources, at a cost of 1000 units to the organisation (500 per resource as stated in Figure 3).

```
<Abuse>
 <Subjects>
   <Subject>ID="123456",Role="student", O="kent.ac.uk"
   </Subject>
 </Subjects>
 <Actions>
  <Action>ID="Get"</Action>
 </Actions>
 <Resources>
  <Resource>ID="www.kent.ac.uk/library"</Resource>
  <Resource>ID="cs.kent.ac.uk/projects"</Resource>
 </Resources>
 <Cost>1000</Cost>
</Abuse>
```

Figure 4. An Example of Abusive Abnormal Behaviour

The Solutions Policy describes the various corrective actions that can be taken, and the cost to the organisation of performing each one of them. Figure 5 shows an example.

```
<SolutionsPolicy>
 <RemoveSubject>
  <ID>Type=Role,Value="Student"</ID>
  <Cost>100</Cost>
 </RemoveSubject>
 <RemoveSubject>
  <ID>Type=Role,Value="Professor"</ID>
  <Cost>1000</Cost>
 </RemoveSubject>
 <UpdateCVP>
  <RemoveAA>LDAPDN="O=Kent,O=AC,C=UK"</RemoveAA>
  <Cost>100000</Cost>
 </UpdateCVP>
 <UpdateCVP>
  <RemoveAtt>Type=Role,Value="Student"</RemoveAtt>
  <Cost>20000</Cost>
 </UpdateCVP>
 <UpdateCVP>
  <RemoveUA>
   <Attribute>Type=Role,Value="Student"</Attribute>
   <Subject>LDAPDN=""</Subject>
   <AA>LDAPDN="O=Glasgow,O=AC,C=UK"</AA>
  </RemoveUA>
  <Cost>2000</Cost>
 </UpdateCVP>
 <UpdateACP>
  <RemovePA>
    <Attribute>Type=Role,Value=Student</Attribute>
    <Action>ID=Get</Action>
    <Resource>ID="www.kent.ac.uk/library"</Resource>
  </RemovePA>
 <Cost>1000</Cost>
 </UpdateACP>
</SolutionsPolicy>
```

Figure 5. An Example Solutions Policy

This policy states that removing a single student user from the system has a cost of 100 units, whereas removing a single professor has a cost of 1000 units. Updating the credential validation policy to completely remove the Kent attribute authority (which means that no credentials issued by Kent will be trusted) costs 100,000 units, whereas completely removing the student role (which means that no students from anywhere will be able to access any resource) has an associated cost to the organisation of 20,000 units. In comparison, removing the user-attribute assignment from Glasgow, so that only its student roles are no longer considered valid, has an associated cost of 2000 units. Updating the access control policy permission attribute assignment for the student role, so that students can no longer Get files from Kent's online library, has an associated cost of 1000 units.

Once the schema for the two policies has been completed, we still will not know how practical or difficult it will be for administrators to set and manage them. The more complex the behavioural rules and solution policies are, the more difficult it will be for administrators to specify all of them. Conversely, if they are too simplistic, they will not be sufficient to control all types of abusive behaviour. Thus significant research is still needed here.

For SAAF to effectively manage a federated RBAC/ABAC authorization infrastructure requires accurate and relevant adaptations against the infrastructure's assets, in light of abnormal behaviour. However, the effectiveness of each adaptation is directly correlated to how well the mechanism for identifying unexpected behaviour operates and the behavioural rules that exist. For example, SAAF can only execute an adaptation as a result of a user breaking rules defined in the behavioural policy. If only a small subset of rules are defined to capture behaviour on critical/sensitive access requests then SAAF will only be able to adapt the infrastructure's assets in relation to those sensitive requests.

It is essential than SAAF's view of the RBAC/ABAC infrastructure, defined by SAAF's asset model, is synchronised with the actual RBAC/ABAC infrastructure. Policies that are currently active in the infrastructure must also be portrayed in the asset model. If a policy changes in the target infrastructure then the asset model must also change. From SAAF's perspective this is maintained through the Executor updating the asset model after every successful adaptation. However this does not cover human interactions, whereby security administrators change active policies without SAAFs knowledge. In a federated environment this becomes even more of a problem, because multiple distributed credential issuing policies are at risk of being changed by many different AA administrators. Therefore a mechanism for monitoring changes in policies must be utilised, with automated updates to SAAF's asset model. This will require the SP to trust the external AAs and give them direct or indirect access to update SAAF's asset model.

We have not yet started implementation. This is the next step. Our plan is to use the PERMIS authorisation system as the first actual RBAC/ABAC system to be controlled. PERMIS contains APIs for accessing and updating all its policies, as well as user attributes and credentials. So SAAF will be able to directly control all of the assets. PERMIS also records the access requests and responses using the XACML request/response context format, so parsing the log records should not be too difficult. We therefore believe that integration with PERMIS will be relatively straightforward. Designing the algorithms for determining abusive behaviour and the appropriate solutions will be more challenging aspects of the research.

4. ACKNOWLEDGMENTS

This research is funded by an EPSRC studentship.

5. REFERENCES

[1] R. Sandu and J. Park, "Usage Control: A Vision for Next Generation Access Control," In Computer Network Security 2776, Springer-Verlag, 2003.

[2] Subramanya, S.R., Yi, B.K. "Digital Rights Management". IEEE Potentials, March-April 2006. Vol.25 Issue 2. pp 31 - 34

[3] The Guardian "WikiLeaks cables: Bradley Manning faces 52 years in jail" http://www.guardian.co.uk/world/2010/nov/30/wikileaks-cables-bradley-manning [accessed: 20 Feb 2011]

[4] C. Bailey, D.W. Chadwick, R. de Lemos, "Self-Adaptive Authorization Framework for Policy Based RBAC/ABAC Models," Proc. 9th Internationl Conference on Dependable, Autonomic and Secure Computing, (DASC 11), 2011, pp. 37-44.

[5] R. L. "Bob" Morgan, Scott Cantor, Steven Carmody, Walter Hoehn, and Ken Klingenstein. "Federated Security: The Shibboleth Approach". Educause Quarterly. Volume 27, Number 4, 2004

[6] H. Debar, M. Dacier and A. Wespi, "Towards a taxonomy of intrusion-detection systems," Computer Networks, 31, Apr 1999, pp. 805-822.

[7] L. Shi and D. W. Chadwick. "A controlled natural language interface for authoring access control policies". Proceedings of the 2011 ACM Symposium on Applied Computing. TaiChung, Taiwan, 21-24 March 2011, pp 1524-1530.

[8] OASIS "eXtensible Access Control Markup Language (XACML) Version 2.0" OASIS Standard, 1 Feb 2005

[9] D.W. Chadwick, G. Zhao, S. Otenko, R. Laborde, L. Su and T.A. Nguyen, "PERMIS: A modular Authorization Infrastructure", Concurrency and Computation: Practice and Experience 20, Aug. 2008, pp. 1341-1357.

Annotating Business Processes with Usage Controls

Andreas Schaad
SAP Research, Security & Trust
Vincenz-Priessnitz Str. 1
76131 Karlsruhe
andreas.schaad@sap.com

Anja Monakva
SAP Research, Security & Trust
Vincenz-Priessnitz Str. 1
76131 Karlsruhe
ganna.monakova@sap.com

ABSTRACT

Complex Supply Chain interactions provide an ideal example of interconnected physical and logical assets that require protection. More specifically, we observe an increasing demand for specifying and enforcing usage control policies within supply chains, relating to both physical + as well as logical assets.

In this paper we will highlight some possible usage control scenarios. We will present our existing control visualization framework to position the identified usage controls in the more general context of safety and security controls. We provide an in-depth discussion of the key constructs of our model and how they can be used to specify and visualize usage controls.

Keywords

Usage Control, Workflow, Security, Visualization

1 Introduction

Consider a typical supply chain between a supermarket, a producer of deep-frozen goods and a logistics provider. Possible usage control scenarios may be to:

- observe a certain temperature during shipment
- not store the shipment next to household cleaning agents
- allow authorized changes of a shipment / purchase order after release
- delete shipment data after completion of shipment
- retain and handle audit relevant shipment data appropriately

Those five scenarios already indicate that we need certain contextual information for specification and later enforcement of usage controls. Equally, we observe that we quite naturally spoke about "the shipment", sometimes referring to a potential physical asset such as a palette, sometimes referring to a logical business object such as a purchase order.

This requires us to consider a conceptual model that would be capable to provide the needed context to define appropriate usage controls as well as an associated execution semantics that can provide support for usage control enforcement.

In this paper we discuss our existing control visualization framework [1] that allows specifying security and safety controls over logical and physical assets. We will then discuss this model in the context of usage controls with a focus on their visualization. A set of possible usage controls will then be analyzed, together with possible mechanisms supporting their specification and enforcement.

2 Control Visualization Framework

Our Control Visualization Framework (CVF) consists of a supply chain risk database; an extended workflow specification language; as well as a workflow execution engine. Our framework explicitly addresses the visualization of safety and security controls on the workflow model and as such we now address the possible integration of usage controls.

2.1 Basic Language Constructs

Figure 1 shows the concepts and their relations used in our supply chain language. A supply chain model is represented by a choreography that contains multiple internal processes represented through activities (hierarchical activities). A choreography specification can contain a number of variables which are basically the representation of the supply chain assets. Variables can be annotated with tags, which identify certain properties of the assets. Each process can have a number of In/Out/InOut arguments, whereby each argument will refer to the variable and therefore to an asset used in the choreography. Output and Input arguments can be connected with a Connector, which specifies the transition of the corresponding asset from one process to another.

2.2 Usage Control Extensions

Overall, our discussion will address how enterprise context can be used for specification of usage control policies at "design-time", as well as how enterprise context can be used at "run-time" to make appropriate usage control decisions. On basis of our simple, yet powerful control visualization model, we now consider its extension with respect to usage controls. We base our discussion along the three core dimensions of usage control [2], namely addressing the data provider and data consumer; provisions and obligations controls; as well as obligation enforcement through signalling and monitoring.

Variables and Tags

Variables essentially describe the assets in our supply chain, and we distinguish between logical assets (such as a purchase order or customer file) and physical assets (such as a physical palette of goods). Tags are then assigned to variables and classify an asset, for example, a purchase order, as audit relevant or the actual shipped good requiring careful handling. This implies that providers will assign the tags to the asset and consumers have to act accordingly when receiving the asset. We, however, now do not only distinguish between data providers and consumers, but rather between asset providers and consumers.

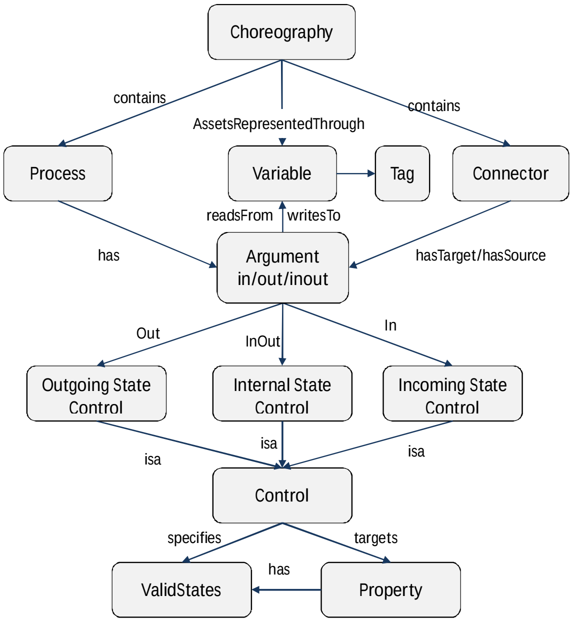


Figure 1: Concepts used in the SCM language

This distinction does not have an impact on provision and obligation controls. We rather observe that in a supply chain context, there are multiple stages where either an asset provider articulates an obligation and where that obligation turns into a provision in a subsequent step of the supply chain.

In other words, the initial step in the supply chain is not necessarily the point in time where all usage controls are defined; this may happen throughout the supply chain. The later enforcement of obligations is driven by the tags and any monitoring needs to be done in accordance with the properties defined for a tag (i.e. frozen goods must be consistently stored at -17 degrees).

Controls and process steps

Controls generically bundle a set of properties, for example, a digital signature control will provide integrity and non-repudiation while a temperature control will provide just temperature. What is important with respect to asset usage is that controls can be enforced at three different states of a process step – input, output, and internal.

This distinction proves to be highly relevant for usage controls, as they relate to the point in time where an asset consumer will accept and then enforce the obligations articulated by the asset provider. For example, if the provider of a logical asset defines that the data in a purchase order must be handled in a confidential manner, then the consumer of this data will check at the input state of the "receive order" process step whether he is able to do this and if so, he will need to enforce this obligation during the internal state of the following "process order" step. Equally, the data consumer will need to restate the obligation at each output state of a process step.

Different control points exist depending on the type of argument. In-arguments can only have input controls that can check the state of the asset before the activity (process) starts its execution; Out-arguments can only have output controls, which check asset state after an activity completed its execution; InOut-arguments can have input, output, as well as internal controls, which control the state of the asset during the activity execution.

2.3 Usage Control Specification Approach

We will now describe how usage controls can be defined together with other safety and security controls in a supply chain. Figure 2 shows the conceptual model of our presented approach. The three main concepts in the model are Asset, Threat and Control. An asset has potential threats and certain controls can countermeasure these threats. The role of the rest of the model is to help identify which threats are applicable to which type of asset and which controls can be used to countermeasure these threats. In the following we describe the steps of our control specification approach based on this conceptual model, specifically focusing on usage controls.

Asset identification

In this step we identify the assets used in a business process that requires controlled execution. As discussed earlier, we identify two types of assets: "Logical assets" representing critical business data such as purchase order details or credit card numbers, while "physical assets" represent real world objects used in the business process, such as a shipment in the supply chain. Any asset can be described by a set of "Properties" it possesses. For example, a logical asset can be described by a set of properties such as signature or encryption properties. Similar, any physical asset can be described by a set of properties such as temperature, location or size. Each property is defined by a set of "states" it can adopt. For example, a temperature property can be in a state −18° C or +5° C, while a signature property can be in the state Unsigned, SignedNoModification, etc.

An asset is characterized by the set of properties it has and the state(s) each property has at a current point in time. This combination between asset type as well as property and state appears to support usage controls. Different types of usage controls can be defined for either a logical or physical asset, but more importantly, we can define what expected properties an asset must exhibit over its lifetime, directly supporting later monitoring and enforcement of usage controls.

Asset classification

It is not sufficient to only distinguish between logical and physical assets, but each asset must be classified. Different threats are applicable to different assets depending on an asset classification. For example, two logical assets can have different threats: the first logical asset might contain private information about a customer with a threat of information disclosure, while another logical asset might contain financial data, which has threat of unauthorized modification. Similar, a frozen physical asset might have threat of being stored at an excessive temperature, while a fragile physical asset may be subject to a threat of being broken. To allow a business process designer to classify business assets, the concept of a "Tag" has been introduced. A tag attached to an asset identifies a certain characteristic or classification of this asset. Figure 4 shows an example set of tags that can be used to classify logical and physical assets. Tags can be attached to the assets in a business process, which would promote awareness of the asset characteristics used in the process.

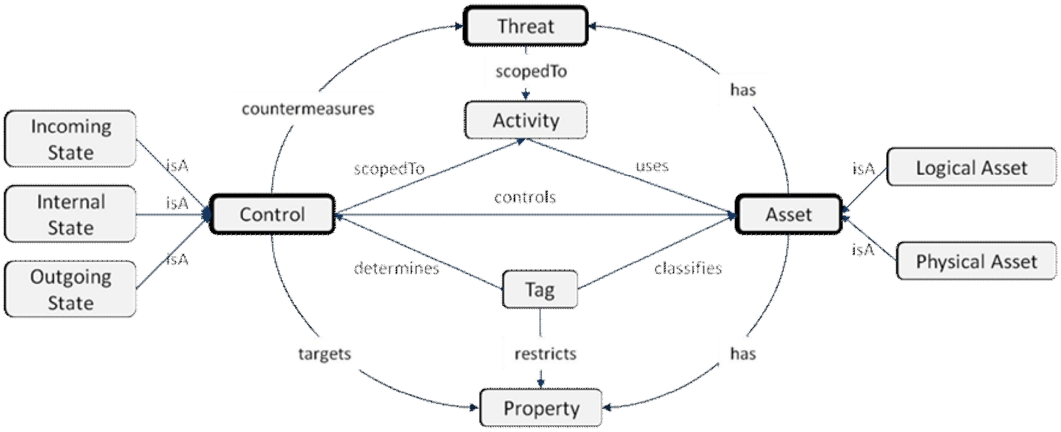


Figure 2: Conceptual Model

In a supply chain example, an Ice Cream asset can be annotated with the tags DeepFrozen and LightSensitive, while a PurchaseOrder can be annotated with the tags AuditRelevant and Financial.

This tagging or classification of assets would have an immediate effect on the definition of later usage controls. For example, if we tag a purchase order as audit relevant, then this would imply a later control over the retention period. Another example could be a customer record asset, tagged as personal information, which would in turn require privacy-aware handling throughout the supply chain process.

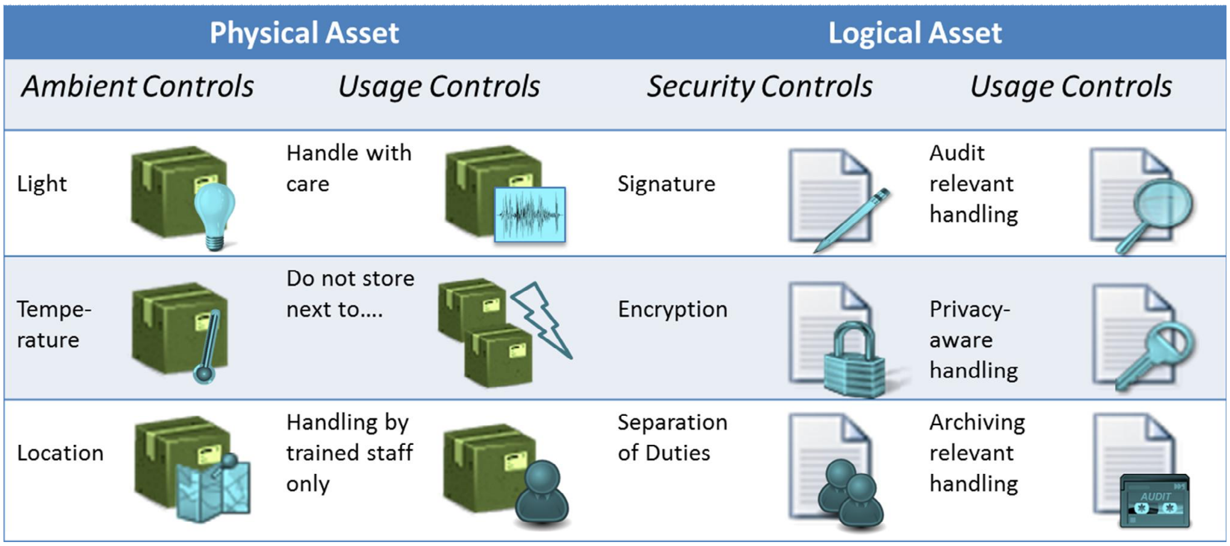
Controls identification

To provide a generic methodology for relating controls to the assets, we classify controls based on the asset properties they can control. For example, a temperature property can be monitored by a temperature sensor control. Similar, a signature property can be controlled by a signature service that can identify whether the document is signed and whether the signature is valid (monitor), or sign the document (enforcer). We distinguish between state-properties and range-properties, and correspondingly state-controls and range-controls. State properties are specified by a list of states a property can take and a state control contains specification of valid states for this property for a given asset at certain time. Range properties are defined by the range of the values it can take, and a range control contains the border specifications for the property values that a certain asset can have at a certain time. Each tag attached to an asset can be viewed as a restriction on certain asset properties. A tag puts restrictions on a property by restricting the set of valid states for this property and the related asset. For example a DeepFrozen tag puts constraints on the temperature property of a physical asset by restricting valid temperature values to under −18° C. Based on tags and implied property restrictions, controls can be identified. For example for the DeepFrozen tag a temperature control will be suggested. To achieve consistency in control identification, the required controls are identified based on the rules stored in a database. The rules derive required controls for each activity that uses an asset annotated with certain tags. Thereby controls can depend on multiple tags as well as on the type of activity that uses the asset. For example, an asset that is flammable and explosive might require different controls than only flammable assets. Controls are implementations of a certain functionality that can control a certain property. A temperature sensor can control a temperature property, while a service that can sign and validate digital signatures is able to control signature property. Figure 3 gives an overview over sample controls for logical and physical assets.

This now again emphasis how our model and reference implementation could handle usage controls at policy specification as well as later runtime. Based on a certain tag (such as audit relevant) and asset type (purchase order) we automatically derive the appropriate usage controls such as guaranteed retention time by deletion only after 10 years.

As mentioned above, controls are related to a certain asset property rather than to an asset, which allows to use the same controls with different assets that have the same property. A signature or encryption service can be used with multiple logical assets, as well as sensors can be used with multiple physical assets. A control “understands” a certain property and can be configured with the valid states for the property and the given asset. The role of the control is to ensure that the asset property the control is responsible for is in a valid state. For example, for a deep-frozen pizza we need controls to ensure that the pizza temperature is under −18° C. Controls are scoped to activities, therefore different activities that use the same assets can have different controls applied to the same assets. Controls can be divided into three main categories – Monitors, Enforcers and Auditors:

- Monitors observe the state of a certain asset property in a specified activity. It can notify a violation in case an invalid state has been detected, display the current states in a dashboard and log them into a database.
- Enforcers are used to transform the state of a property. For example, a signature property enforcer can automatically sign created documents, while a temperature property enforcer might be able to switch on an emergency freezer if the temperature monitor detects that the current temperature is too high.



Physical Asset		Logical Asset	
Ambient Controls	*Usage Controls*	*Security Controls*	*Usage Controls*
Light	Handle with care	Signature	Audit relevant handling
Temperature	Do not store next to….	Encryption	Privacy-aware handling
Location	Handling by trained staff only	Separation of Duties	Archiving relevant handling

Figure 3: Example Controls & Visualizations

- Auditors generate reports on property state history. Auditors use data logged in by the monitors to compute specified functions. For example, an auditor can analyze if the temperature of an asset was above limit for longer than 5 minutes and correlate this information with the asset location.

Control points identification

A control can be applied at different stages of an activity execution. If applied on activity initialization, it can control the incoming states of the asset properties; if applied on activity execution, it can control the internal states of the asset properties; if applied on activity completion, it can control outgoing states of the asset properties. Depending on the type of activity, different control types are applicable. Incoming state controls and outgoing state controls can be enforced by the workflow engine – it can invoke control services to verify that the asset properties are in a correct states and can for example suspend a workflow (or execute any other activities that are defined as part of a reactive process) if a violation has been detected. The internal controls on the other side can be viewed as the requirements on the activity implementation with regard to the asset handling. Having such requirements as part of the model can be used for example for generation of contracts between participants from the designed process model. The next section describes how the presented approach has been realized in a prototype.

3 Architecture

Figure 4 gives an overview over the architecture in a SOA environment. At the design time, the RiskDB is consulted to identify threats and countermeasures for the business process assets that have been classified with the tags. At the runtime, process execution engine invokes control services at the specified control points through the control service broker. All controls are available as property control services that subscribe to the property they can control in the RiskDB, specifying the type of the control (monitor, enforcer, or assessor) and the assets it can handle. A business process engine sends the asset or asset reference and the property to control to the control service broker, which then looks up available services in the RiskDB and finds a service that can evaluate or change the state of the given property for the given asset. For example, a sensor platform will find the sensor that is attached to the given asset, and a signature service suitable for the given document type will be selected. All property states, as well as process execution states are stored in a LogDB, which feeds data into the dashboard and allows offline analysis of the completed instances and improvement of the rules specified in RiskDB.

4 Implementation

Our current prototype is based on Windows Workflow Foundation (WF 4.0). Figure 4 shows the prototype architecture, where the bold elements represent our extensions to the WF4.0 framework.

Microsoft Workflow Foundation uses variables to represent data used in a business process, however, the variables are defined in a variable tab and are not visible in the designer. To advocate security awareness, we extended existing workflow modelling constructs with two visual elements for logical and physical assets. Furthermore, we added an asset (or variable) panel to the business process, which contains all assets used in the process. To add a new asset (variable) to the process, the user just needs to drag & drop the corresponding visual element into the asset/variable panel of the workflow. To enable asset classification we provided a tag toolbar. To annotate a variable the user needs to drag & drop the corresponding tag from the toolbar onto the visual asset specification now present in the asset panel. By combining different tags, a user can specify different characteristics of an asset. Figure 5 shows a screenshot of the ice cream supply chain process modelled using our tool. It contains two variables that can be seen in the right panel: An IceCream variable annotated with a DeepFrozen and LighSensitive tags and a PurchaseOrder variable annotated with Financial and AuditRelevant tags. On basis of these tags we would now define the possibly required usage controls.

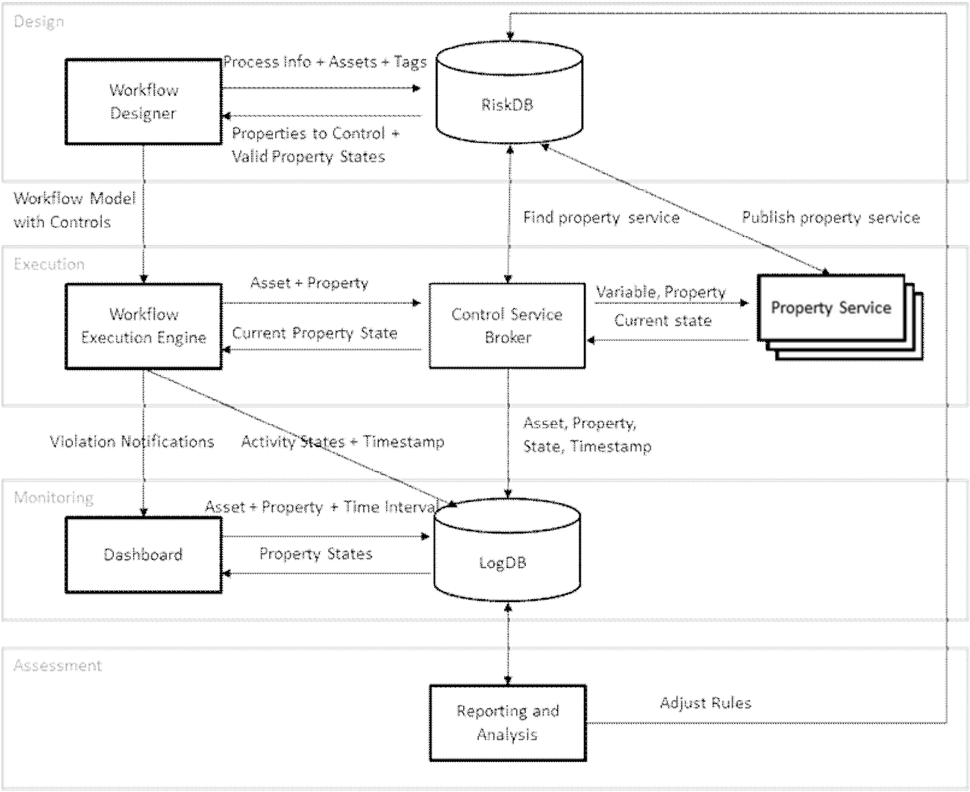


Figure 4: Architecture

In Figure 5 we can see four activities: Order, Dispatch, Transport, and Receive activities. The Activity Order outputs PurchaseOrder, which is then passed as an input argument to the Dispatch activity. The Dispatch activity then outputs IceCream, which is passed to Transport activity and then through the Transport activity to the Receive activity. Depending on the type of argument (In, Out or InOut), we can see different types of control points available for each asset in each activity. This allows the user to define input state controls on the incoming asset states (PurchaseOrder in Dispatch activity) output controls on outgoing asset states (PurchaseOrder in Order activity), and internal controls on data that exists all the way through activity execution (IceCream in Transport activity). These would be the points in the supply chain execution where usage controls would be enforced and monitored.

To identify controls required to countermeasure potential threats or usage control requirements, we developed a Risk Database (RiskDB). The RiskDB stores relations between asset tags, threats these tags imply for different activities, and controls that should be applied to such assets in each activity. When a user annotates an asset with a new tag, a query is sent to the RiskDB that selects the necessary protection measurements (or controls) for each activity that uses this asset. After this the tool checks if the controls are already present in the model and if not, shows an error with the information about missing controls. This requires a business designer to model secure processes with respect to the rules stored in the RiskDB.

To enable control specification, we provide a control toolbar. To identify at which point of activity execution a control must be applied, the user needs to drop a control into the corresponding container. In Figure 5 we can see an output signature control applied to the PurchaseOrder variable in Order activity. This control specifies that the data must be signed when it leaves this activity. In the Dispatch activity we can see an example incoming state control that states that the PurchaseOrder signature property must be in state verified to be used by this activity. In the Transport activity internal temperature and light controls are specified, which define that IceCream temperature must be between −50°C and −25°C and light must be under 200 Lumen. Additional controls could be added as input and output controls.

In general, any number of controls can be applied to each asset in each activity. For example, a possible usage control on the Purchase Order asset could be that the supermarket chain ordering the ice cream asks the icecream manufacturer to not share any non-relevant details of the order (eg price) with the logistics provider. This would then imply that at execution time, the purchase order file is sanitized, ie the usage control would be placed on the purchase order asset at the outgoing asset state.

Another example could be a usage control demanding that the logistics provider deletes all shipment data after 60 days. In this case, we would place a control on the outgoing purchase order asset state in the Transport activity which would eventually trigger a timed deletion event.

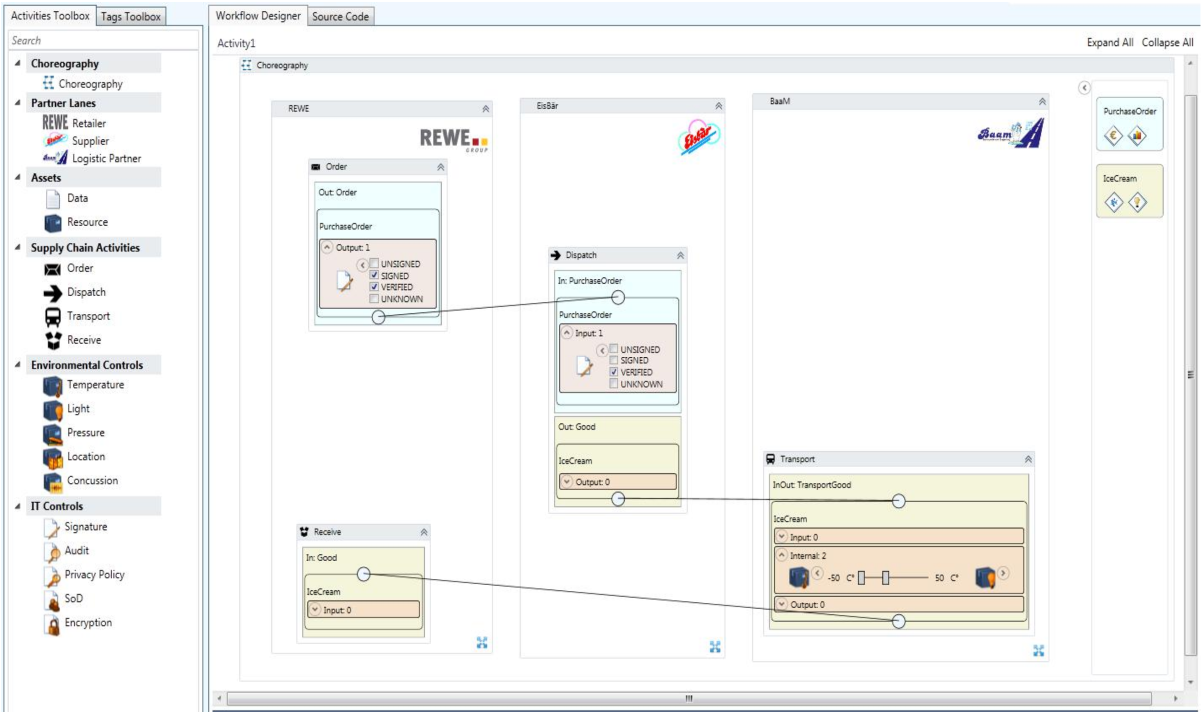


Figure 5: Screenshot of a Modelled Supply Chain & Example of Applied Controls

5 SUMMARY AND CONCLUSION

In this paper we discussed our existing control specification framework [2] and its possible extension in the context of usage controls. One key finding was that we can equally specify usage controls on physical as well as logical assets. Tagging assets in our supply chain does allow automatically inferring appropriate controls and then enforcing them at workflow execution time. We demonstrated how we envision the later visualization of usage controls.

Of course many points remain open and require further research, though we consider them to be outside the scope of this paper. For example, while certain controls are quite straight forward to automatically implement (eg. a simple digital signature) other controls appear to require more contextual information, both at specification and runtime, and we need to consider a possible application of our earlier transformation approaches [3].

Another point is further required work on more fine-grained usage control taxonomies as there appears to be no existing work on basis of which we could provide a more detailed visualization of controls. The presented visualizations in this paper are of course rudimentary and would require involvement of the HCI community such as seen in earlier SOUPS workshops. We however envision that next generation UI framework such as HTML 5 or MS WPF will allow definition of more interactive (usage control) policy Widgets. For example, we could consider widgets that actually incorporate selection boxes, pull-down menus or input fields.

Future work will now look into associating specific usage control mechanisms to our business process and control visualization platform. The PrimeLife policy engine [4] should allow us to specify and then enforce privacy-specific usage control policies. Sanitizable signature schemes [5] could support allowed modification of signatures depending on intended usage and supply chain state. Provable data possession schemes [6] could be used to articulate usage control requirements such as “only process order if you have obtained a safety clearance”.

REFERENCES

[1] Ganna Monakova, Achim D. Brucker and Andreas Schaad. Security and Safety of Assets in Business Processes. In ACM Symposium on Applied Computing (SAC), ACM Press, 2012

[2] Hilty, M., Pretschner, A., Basin, D., Schaefer, C., Walter, T.: A Policy Language for Usage Control. 12th European Symp. on Research in Computer Security (ESORICS), pp. 531-546, Dresden, September 2007

[3] Christian Wolter, Michael Menzel, Andreas Schaad, Philip Miseldine, Christoph Meinel: Model-driven business process security requirement specification. Journal of Systems Architecture - Embedded Systems Design 55(4): 211-223 (2009)

[4] Slim Trabelsi, Jakub Sendor, Stefanie Reinicke: PPL: PrimeLife Privacy Policy Engine. POLICY 2011:184-185

[5] G. Ateniese, D. Chou, B. de Medeiros, and G. Tsudik. Sanitizable signatures. In ESORICS’05, 2005.

[6] Giuseppe Ateniese, Randal C. Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary N. J. Peterson, Dawn Xiaodong Song: Provable data possession at untrusted stores. ACM Conference on Computer and Communications Security 2007: 598-60

An Approach to Data-driven Detective Internal Controls for Process-aware Information Systems

Rafael Accorsi
University of Freiburg, Germany
accorsi@iig.uni-freiburg.de

ABSTRACT

This paper argues for an approach for the well-founded, scalable detective internal controls to assist controllers in swiftly and reliably identifying violations of control objectives in business process executions. Considering the usual internal control setting, in which controllers have a process and policy specification (target state) and the corresponding event log generated during the process execution (actual state), our approach automatically analyzes the entire set of process executions comprised in the event log. For this, novel, formal approaches to data-driven conformance checking need to be devised.

Categories and Subject Descriptors

K.6.5 [**Manage-ment of Computing and Information Systems**]: Security and Protection

Keywords

Detective internal control, Business Process Management, Usage control

1. INTRODUCTION

Detective controls are designed to identify, a posteriori, the violation of control objectives in enterprise information systems. Control objectives include, for instance, abuse of rights, conflict of interest and four-eye rule. Generally, such controls, as well as compliance rules, can be regarded as usage control requirements [3, 23, 26].

Despite the recent series of accounting failures and associated regulation efforts, detective internal control practices for business processes – and more generally process-aware information systems [19] – are still based upon the manual analysis of random sample logged process executions [36]. The resultant control risk is high, i.e. the probability and the associated costs of overlooking violations, thereby endorsing fraud or eventually failing an audit.

The Société Génerale incident is a particularly prominent, well-documented example to illustrate the impact of flawed internal controls. Unauthorized transitions by trader Jérôme Kerviel led to the loss of nearly 5 billion Euros. These transactions (e.g., directional bets concealed by fake portfolios) were only possible because internal controls, such as those for segregation of duties and abuse of rights, have been circumvented. These errors were not spotted despite the availability of complete logs, whereas the reason for this is faulty detective internal controls [20].

This paper argues for an approach for automated data-driven detective internal controls. We call it Adict. Adict builds upon conformance checking, i.e. analysis based upon comparisons between the target state (process specification) and the actual state (event logs). Conformance checking [34] is a technique within the field of process mining [33], which is employed to detect discrepancies between the target and the actual behavior. However, up to now only structural features of process runs, such as deviating executions and incidence of paths, could be detected. The consideration of more sophisticated control objectives or security policies (e.g., separation of duties and usage control requirements) are not possible.

Adict extends conformance checking to determine the compliance of event logs with various types of control objectives. Based upon colored Petri nets, which provide a suitable semantics for data-driven business process reasoning [8], the techniques to be developed in Adict address: (a) the declarative formalization of process-independent, semantically justified control objectives as Petri net anti-patterns, whereas specific places in these patterns denote control objective violations; (b) the analysis of process executions by replaying the event log traces in the process specification to detect violations of the patterns; and (c) the automated derivation and circumscription of need-to-know requirements based upon the target and the actual states.

The remainder of this paper is structured as follows. After a brief survey on related approaches, Section 2 gives an overview of Adict's approach and its main building blocks and Section 3 and indicates ongoing work.

Web as a more general application context.

While the techniques suggested in this paper are motivated by and shown in the context of internal control/auditing, we believe that they could be equally employed in other settings, in particular the web. In essence, Adict is an approach for detective usage control, i.e. a posteriori generation of compliance evidence with the designated policies.

DUMW'11 Lyon, France.

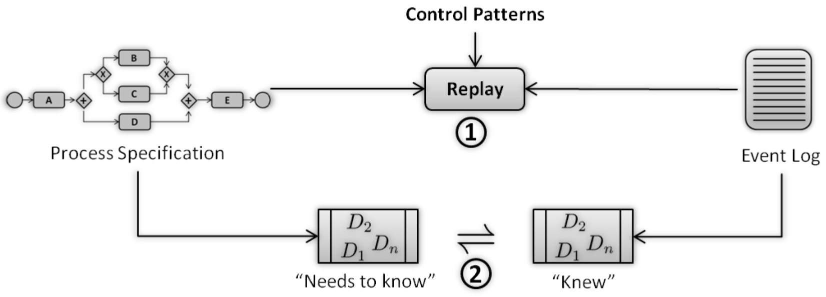


Figure 1: Overview of Adict.

One exemplary application could be a "Social network dashboard". Here, Adict could provide users with pre-defined patterns capturing usage control policies (e.g., on data retention and third-party notification), so that users could click their way through the policies. Upon request, a particular view of the log – corresponding to the user – could be generated and checked for compliance.

The realization of such a feature anticipates, one the one hand, the willingness of service providers to make their processes (at least in part) public. This could be achieved in situations where economical incentives to transparency were in place. On the other hand, it assumes a reliable infrastructure that includes secure logging and remote attestation technologies [2]. Otherwise the evidence generated during the check is void.

Related work.

We have amply investigated the use of conformance checking to conduct security and compliance audits in process-aware information systems [6]. This case study indicates that there are mechanisms to determine which traces in the event log fit into the model, as well as to perform straightforward compliance checks based solely upon the log files [1]. However, checking data-driven constraints, such as data propagation, is not possible. This appears to be a more general shortcoming, and in fact challenge, when it comes to analyzing log data "in the large" [24] and process mining.

Aalst et al. [35] present the Online Auditing Tool (OLAT) framework which in essence integrates the existing mechanisms to provide for continuous auditing and, in further stages, also preventive monitoring. However, in OLAT the actual state is not employed. In addition, only elementary control objectives can be considered. Accorsi et al. [4] develop a monitor architecture based upon conformance checking to identify and evaluate process deviations during its execution with regard to the compliance with security rules.

Some approaches to detective internal control rely solely upon event logs (e.g., [5, 13, 17]). They cannot be seen as conformance checking in the strict sense, as there is no consideration of the target state (in terms of process model) during the analysis.

2. APPROACH OVERVIEW

Figure 1 provides an overview of the Adict approach. It builds upon two types of conformance checking. Firstly, replays determine whether control objectives formalized as Petri net patterns, are violated. For this, a Petri net representation of the process is employed, on top of which traces are replayed. Similar to security automaton [29], whenever a replayed trace activates a pattern, the corresponding control is violated. (Note that patterns capturing the violation of a property are generally referred to as anti-patterns. For the sake of readability, below we simply refer to them as "patterns".) Secondly, Adict focuses on need to know requirements, which while relevant, are usually not considered or checked automatically. It employs abstractions to circumscribe, from the target state, the data set each subject needs to know (as for the process specification) and derives from the event log the data set knew by each subject; subsequent conformance checks with the data set "needed to know" and the data set "allowed to know" provide evidence on violations of abuse of rights (and hence "bad" information flows) and improper policy specification.

2.1 Business Process Specification

Business processes are traditionally specified using languages such as the Business Process Modeling and Notation (BPMN), Business Process Execution Language (BPEL) and Event-driven Process Chains (EPC). Generally, they have an execution semantics, but lack a formal semantics to allow the automated reasoning. The usual way to circumvent this problem is to map the specifications to Petri net models [32]. (Alternatively, process algebra can be employed. Still, the vast majority of approaches employ Petri nets.) Here, the process activities are considered transitions of the net and performing a transition consumes a token from one place, and produces the corresponding tokens on the other. The most notable dialect for this purpose is the Workflow net [32], which restricts the models regarding the form and transition semantics; for instance, that the net has unique and distinct start and end places, and that the black tokens are completely passed over during the execution.

However, Workflow nets do not allow for the representation of data items (and, more generally, resources). Correspondingly, approaches to mapping business process specifications to Petri nets consider only the structure and control flow of the process, not the exchanged data items. As a preparation for Adict, we devised a more expressive formalism called Information Flow Net (IFnet) [7]. IFnet combines colored Petri nets and Workflow Nets, and define mappings from BPEL and BPMN into IFnet models. (Definitions of soundness and the corresponding decidability results, adapted from Workflow nets, are available and hold.)

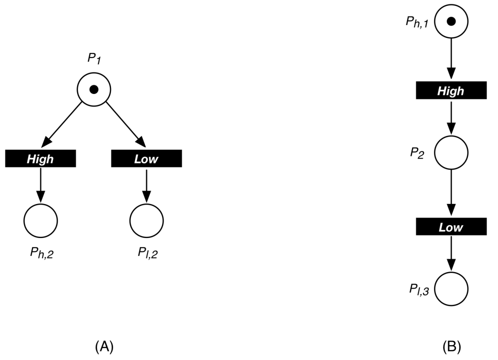


Figure 2: Patterns to capture interferences.

In doing so, Adict is able to reason about such data items, for instance, whether a data item moved down from a secret to a public domain.

2.2 Characterization of Control Objectives

Typical control objectives in process-aware information systems are [31]:

- *Four-eye principle*: business decisions and transactions need approval from two distinguished subjects prior to commitment.
- *Segregation of duties*: dissemination of activities and associated privileges for a specific business process among multiple subjects.
- *Binding of duties*: assignment of activities and associated privileges for a specific business process to one subject.
- *Conflict of interest*: subjects (and information) involved in the execution of one process should not be involved in the execution of another process.
- *Need-to-know*: subjects should only obtain the information necessary to run a specific process or carry out a particular task.

Generally, control objectives also comprise data usage requirements [12]; for example, that a data must be deleted after the execution of a process (data flow), or that an activity must be isolated from another set of activities (interference). Further, regulatory compliance requirements can be reduced to usage control requirements [26, 27] and, hence, be equally seen as control objectives.

Adict captures these control objectives as IFnet patterns in a way similar to [10]. In previous work, we employed such a patterns to provably capture particular information flow properties, in particular mandatory access control rules and different kinds of interferences, thereby extending "Place-based Non-Interference" [16]. To exemplify this specification style, the patterns in Fig. 2 capture a specific kind of non-interference, i.e. covert information flows. Specifically: assuming a multi-level security model [18], the patterns in Fig. 2 capture the Bisimulation-based Non-Deducibility on Composition (BNDC), which forbids a low subject from deriving information about high's behavior. At the conflict place P_1 in Fig. 2(a), high and low compete for the black token (control flow) and whenever high consumes the token, low can deduce high's action. At the causal place P_2 in Fig. 2(b) one action of high always follows one action of low. Overall, the control flow allows low to deduce information about high (interference), thereby violating, e.g., isolation.

For the moment, Adict has patterns to capture, for example, separation (and thus binding) of duties, conflict of interest and data flow requirements based upon mandatory access control.

2.3 Log Format

The log-based analysis of business process is generally referred to as process mining [33]. Process mining encompasses three types of approaches: *discovery* to reconstruct so-called de-facto process models from logs; *conformance* to check the extent to which the logs correspond to the original de-jure process; and based on such analysis, *enhancement* to improve the model in order to fulfill the expected properties. Adict focuses on conformance checking, as it builds upon comparing the actual and the target states.

The starting point for conformance checking is an event log. Each event in such a log refers to an activity (a well-defined step in some process) and is related to a particular case (a process instance). The events in a case are ordered and describe one "run" or "trace" of the process. Event logs also store supplementary information, such as the originator (person or device) triggering the activity, its role, the event's time stamp, required input and provided output. The following depicts the typical log format as input for process mining.

timestamp	activity	originator	input data	output data

The key assumption here is that the designated process-aware information systems provide for these fields, or that the corresponding log formats can at least be reduced to the format. There is enough evidence to substantiate this assumption [37]. Log formats, such as the eXtensible Event Stream (XES), allow for the realization of efficient mechanisms for log analysis, for instance process discovery and conformance checking.

2.4 Conformance Checking

In conformance checking, an existing process model is compared with an event log of the same process [34]. The comparison shows where the real (executed) process deviates from the modeled process. Moreover, it is possible to quantify the level of conformance and differences can be diagnosed. Conformance checking can be used to check if reality, as recorded in the log, conforms to the model and vice versa. There are various applications for this (compliance checking, auditing, Six Sigma, etc.). Adict exploits conformance checking for detective internal controls, which is in itself similar to an auditing setting (log analysis), even though under a different set of assumptions [31].

Conformance checking and performance analysis require an alignment of event log and process model, i.e., events in the event log need to be related to model elements and vice versa. Such an alignment shows how the event log corresponds to the process model. Assuming a business process specification and an event log as in Section 2.3, the central mechanism to check their conformance (or alignment) consists of replaying the activities in the log into the corresponding activities of the process specification. Replay thus detects structural discrepancies between the target model and logs, in that traces, for example, execute activities in the wrong sequence or skip required transactions. Technically, considering a Petri net representation of the process, replay is realized as a Petri net "token game" [22] by forcing transitions to fire (if possible) in the order indicated by the trace.

Currently, conformance checks based upon replay address only structural aspects of the workflow. That is, data-driven checks (e.g., encompassing message passing or data exchange) is are possible; similarly, currently it is not possible to make the originator accountable for a certain task, as this information (log field) is not employed by the checks. Adict extends conformance checking with these dimensions. Furthermore, while triggering the activities, it also triggers the corresponding activities in the patterns that capture the control objectives. In doing so, the corresponding tokens are moved on in the control objectives patterns and, if applicable, indicate a violation whenever a "harmful" place is active.

2.5 Addressing Need to Know Requirements

The principle of "need to know" restricts the set of information that can be known by a subject to those data items strictly necessary to conduct the designated duties in a process. Need to know is closely related to the principle of "least privilege" [28] and often equated with it [15, 30]. This principle suggests that each subject in a system should be granted the most restrictive set of privileges (or the lowest "clearance") needed for the performance of authorized tasks.

Although related, least privilege and need to know focus on different aspects and, hence, require different mechanisms. While the former focuses on the rights, the latter focuses on the (maximal) set of data which can be accessed by a subject. Because in enterprises and corresponding process-aware information systems roles (and thereby subjects) usually possess more rights than those needed for the execution of a particular process [14], cascading accesses may take place [11], vulnerabilities exist [21] and break-glass policies allow for the temporary elevation of rights [25], detective internal controls must check whether these situations led to an abuse of rights in which a subject obtains more information than possible. Further, they should indicate whether covert access may have led to information gain. Put another way: rather than focusing solely on the rights, need to know must focus on the information that potentially flows to the subject.

Adict employs abstraction techniques to characterize, based upon the process specification, the set of data a subject "needs" to know in order to conduct a process. Similarly, abstraction, will be employed to obtain the set of information that such a subject "knew", together with possible interferences (Section 2.2). This allows Adict to detect discrepancies between these two "epistemic" states and, consequently, identify violations of control objectives, as well as abuse of rights.

3. SUMMARY

This paper argues for several types of conformance checking for the automatic detective internal controls. The goal is to improve the quality of process-aware information systems by reliably and timely detecting violations and, thereby, allow the enhancement of process design (or execution engine). The Adict approach provides for declarative, process-independent characterizations of control objectives that can be straightforwardly mapped to process-specific patterns and, subsequently, serve as a basis for conformance checking. Given that, replays attempt to reproduce the traces into the model, simultaneously triggering the corresponding patterns. Further, Adict addresses the characterization and detection of need to know requirements in the context of business processes. To the best of our knowledge, this is a novel, promising of reasoning about automated detective controls. We have carried out experiments with a prototypical implementation focusing solely on properties encoded on solely structural patterns. (To this end, we synthesized log files with SWAT, the Security Workflow Analysis Toolkit [9].) While the Python implementation detect all the violations in the log, it took around a minute to traverse an event log with 500K cases. (We employed in the test a virtual machine with Ubuntu 10.10 64-Bit, 4GB RAM and one core with 2,67 GHz). Further optimizations are possible.

A particular attractive feature on Adict is that it allows the quantification of incidents. The fact that the log "chunk" designates the particular view of the reality that held for a time period makes it possible to determine, for the designated time period, for instance the amount of information that flows over a covert-channel. We will exploit this dimension in a future time point.

4. REFERENCES

[1] R. Accorsi. Automated privacy audits to complement the notion of control for identity management. In E. de Leeuw, S. Fischer-Hübner, J. Tseng, and J. Borking, editors, *Policies and Research in Identity Management*, volume 261 of *IFIP Conference Proceedings*, pages 39–48. Springer, 2008.

[2] R. Accorsi. Log data as digital evidence: What secure logging protocols have to offer? In *Proceedings of the 1st IEEE Workshop on Computer Forensics in Software Engineering*, pages 398–403. IEEE Computer Society, 2009.

[3] R. Accorsi, L. Lowis, and Y. Sato. Automatisierte compliance-zertifizierung cloud-basierter geschäftsprozesse. *Wirtschaftsinformatik*, 53(3):139–149, 2011.

[4] R. Accorsi, Y. Sato, and S. Kai. Compliance monitor for early warning risk determination. *Wirtschaftsinformatik*, 50(5):375–382, October 2008.

[5] R. Accorsi and T. Stocker. Automated privacy audits based on pruning of log data. In *Proceedings of the EDOC International Workshop on Security and Privacy in Enterprise Computing*. IEEE, 2008.

[6] R. Accorsi and T. Stocker. On the exploitation of process mining for security audits: The conformance checking case. In *ACM Symposium on Applied Computing*, 2012.

[7] R. Accorsi and C. Wonnemann. Strong non-leak guarantees for workflow models. In *ACM Symposium on Applied Computing*, pages 308–314. ACM, 2011.

[8] R. Accorsi and C. Wonnemann. InDico: Information flow analysis of business processes for confidentiality requirements. In J. C. et al., editor, *ERCIM Workshop on Security and Trust Management*, volume 6710 of

Lecture Notes in Computer Science, pages 194–209. Springer, 2011.

[9] R. Accorsi, C. Wonnemann, and S. Dochow. SWAT: A security workflow toolkit for reliably secure process-aware information systems. In *Conference on Availability, Reliability and Security*, pages 692–697. IEEE, 2011.

[10] N. Adam, V. Atluri, and W.-K. Huang. Modeling and analysis of workflows using petri nets. *Journal of Intelligent Information Systems*, 10(2):131–158, 1998.

[11] R. Anderson. *Security Engineering*. Wiley, 2nd edition, 2008.

[12] V. Atluri and J. Warner. Security for workflow systems. In M. Gertz and S. Jajodia, editors, *Handbook of Database Security*, pages 213–230. Springer, 2008.

[13] D. A. Basin, M. Harvan, F. Klaedtke, and E. Zalinescu. Monitoring usage-control policies in distributed systems. In C. Combi, M. Leucker, and F. Wolter, editors, *Symposium on Temporal Representation and Reasoning*, pages 88–95. IEEE, 2011.

[14] M. Benantar. *Access Control Systems*. Springer, 2006.

[15] J. Biskup. *Security in Computing Systems - Challenges, Approaches and Solutions*. Springer, 2009.

[16] N. Busi and R. Gorrieri. Structural non-interference in elementary and trace nets. *Mathematical Structures in Computer Science*, 19(6):1065–1090, 2009.

[17] J. Cederquist, R. Corin, M. Dekker, S. Etalle, J. den Hartog, and G. Lenzini. Audit-based compliance control. *International Journal of Information Security*, 6(2-3):133–151, 2007.

[18] D. Denning. A lattice model of secure information flow. *Communications of the ACM*, 19(5):236–243, 1976.

[19] M. Dumas, W. van der Aalst, and A. ter Hofstede, editors. *Process-aware Information Systems: Bridging Poeple and Software through Process Technology*. Wiley, 2005.

[20] J. Epstein. Security lessons learned from Société Générale. *IEEE Security & Privacy*, 6(3):80–82, 2008.

[21] L. Lowis and R. Accorsi. Finding vulnerabilities in SOA-based business processes. *IEEE Transactions on Service Computing*, 4(3):230–242, 2011.

[22] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[23] K. Namiri and N. Stojanovic. Using control patterns in business processes compliance. In M. Weske, M.-S. Hacid, and C. Godart, editors, *Proceedings of the International Workshop on Web Information Systems Engineering*, volume 4832 of *Lecture Notes in Computer Science*, pages 178–190. Springer, 2007.

[24] A. J. Oliner, A. Ganapathi, and W. Xu. Advances and challenges in log analysis. *Communications of the ACM*, 55(2):55–61, 2012.

[25] D. Povey. Optimistic security: A new access control paradigm. In *Proceedings of the New Security Paradigm Workshop*, pages 40–45. ACM Press, 1999.

[26] A. Pretschner, F. Massacci, and M. Hilty. Usage control in service-oriented architectures. In C. Lambrinoudakis, G. Pernul, and A. M. Tjoa, editors, *Proceedings of the 4th International Conference on Trust, Privacy and Security in Digital Business*, volume 4657 of *Lecture Notes in Computer Science*, pages 83–93. Springer, 2007.

[27] S. Sackmann and M. Kähmer. ExPDT: A policy-based approach for automating compliance. *Wirtschaftsinformatik*, 50(5):366–374, October 2008.

[28] J. Saltzer and M. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, September 1975.

[29] F. Schneider. Enforceable security policies. *ACM Transactions on Information and System Security*, 3(1):30–50, February 2000.

[30] F. B. Schneider. Least privilege and more. *IEEE Security & Privacy*, 1(5):55–59, 2003.

[31] E. Vaassen, R. Meuwissen, and C. Schelleman. *Accounting Information Systems and Internal Control*. Wiley, 2010.

[32] W. van der Aalst. The application of petri nets to workflow management. *Journal of Circuits, Systems, and Computers*, 8(1):21–66, 1998.

[33] W. van der Aalst. *Process Mining – Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.

[34] W. van der Aalst, A. Adriansyah, and B. van Dongen. Replaying history on process models for conformance checking and performance analysis. *Data Mining and Knowledge Discovery*, 2012.

[35] W. van der Aalst, K. van Hee, J. M. van der Werf, A. Kumar, and M. Verdonk. Conceptual model for online auditing. *Decision Support Systems*, 50(3):636–647, 2011.

[36] W. van der Aalst, K. van Hee, J. M. van der Werf, and M. Verdonk. Auditing 2.0: Using process mining to support tomorrow's auditor. *IEEE Computer*, 43(3):90–93, 2010.

[37] H. M. Verbeek, C. A. M. Joos, B. Dongen, and W. van der Aalst. XES, XESame, and ProM 6. In W. Aalst, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, C. Szyperski, P. Soffer, and E. Proper, editors, *Information Systems Evolution*, volume 72 of *Lecture Notes in Business Information Processing*, pages 60–75. Springer, 2011.

A Policy Driven Semantic Approach to Data Usage Management

Anupam Joshi, Tim Finin, Karuna Joshi, Madan Oberoi
CSEE Department, UMBC
Baltimore, MD 21250
{joshi, finin, kjoshi1}@umbc.edu

Abstract— **As the amount of information available on the web has increased, several privacy and security issues around the use of such information have arisen. Government (and private) entities are able to gather and analyze data from several disparate sources with ease. This ability to do large scale analytics of publicly accessible data leads to significant privacy concerns, especially when done by governments. The converse is also true, with concerns about data being shared by individuals and organizations to the web and the cloud. Our work develops a semantically rich, policy driven approach to address the privacy, security and usage concerns around such data.**

Introduction and Motivation

In today's highly networked information infrastructure, a significant amount of information is accessible publicly over the web. Such information is gathered by a variety of government and private entities. This information, gathered from a variety of sites, can be linked together and analyzed to make inferences about entities of interest. While the expectations of privacy vary with culture and country, it appears that often citizens are relatively more comfortable with commercial companies mining their personal information rather than law enforcement agencies collecting and mining this data across information sources. One concern in particular is that Law Enforcement or Counter Intelligence agencies often use such public information to "fish" for potential suspects [1, 2, 3]. Similar concerns about data aggregation have also been expressed recently about companies (such as Google, Facebook, etc) that provide a platform with a variety of applications that are commonly used.

A related issue is the problems being faced by cloud/web based service platforms. These have the promise to significantly lower the cost and increase the effectiveness of many data storage, access, and analysis tasks. However, reluctance of individuals and organizations to share data because of privacy, confidentiality, and usage concerns is preventing their adoption. Within the past year for instance, the federal government in the US has mandated that data centers be consolidated, and that a set fraction of the federal IT tasks be done using (public) clouds [32]. A key barrier to this however is the reluctance of the CIOs to let data go outside the organization, since they cannot ensure that the cloud/web based provider will be able to meet the organization, as well as legal/statutory constraints on sharing and usage that they have to enforce.

Our research has sought to address this issue by using machine understandable and semantically rich descriptions of the a) data, b) policies governing access, usage and privacy, and c) the query context

Related Work

The TAMI (Transparent Accountable Data-mining Initiative) project attempts to address issues of transparency, accountability in context of personal privacy by changing the perspective from controlling or preventing access to encouraging appropriate use of accessed data and inferring when data is misused by investigating the audit logs [10]. Our proposed work is closely related as it relies on logs to figure out whether obligations are met. However, unlike TAMI, our model does enforce privacy policies but does so on the end use data produced as a result of the query instead of the initial data dump required.

Kagal, Hanson and Weitzner [11] have discussed providing explanations associated with the derivation of a policy decision in the form of a list of reasons, called dependencies by them, using semantic web technologies. This kind of explanations will help the user as well as database owner agencies to understand how the results were obtained, thereby increasing trust in the policy decision and enforcement process. Our model will provide similar justifications about query decisions.

A lot of work has been done to develop machine interpretable policy frameworks [12], [13]. Rein (Rei and N3) [14] is a distributed framework for describing and reasoning over policies in the Semantic Web. It supports N3 rules [15], [16] for representing interconnections between policies and resources and uses the CWM forward-chaining reasoning engine [17], to provide distributed reasoning capability over policy networks. AIR [18] is a policy language that provides automated justification support by tracking dependencies during the reasoning process. It uses Truth Maintenance System [19] to track dependencies. Policies and data are represented in Turtle [20], whereas the reasoning engine is a production rule system [21] with additional features for improved reasoning efficiency such as goal direction. Rei and AIR consider rules defined over attributes of classes in the domain including users, resources, and the context. Though our initial prototype uses OWL to describe privacy policies, we plan to use AIR in the future to take advantage of its built-in justification feature.

Letouzey et al [22] have discussed existing security models by defining the security policy through logically distributing RDF data into SPARQL views and then defining dynamic security rules, depending on the context, regulating SPARQL access to views. Kagal and Pato [23] have explored the use of semantic privacy policies, justifications for data requests, and automated auditing to tackle the privacy concerns in sharing of sensitive data. Their architecture evaluates incoming queries against semantic policies and also provides a justification for permitting or denying access, which helps requesters formulate privacy-aware queries. Currently our conceptual model does not restrict the query language to be used, but we plan to use SPARQL for better integration with Semantic Web data sources.

FRAMEWORK

The basis of our approach is the use of policies that describe the data, along with the constraints on that data (who can access it, under what circumstances, for what use etc.) that the individual or the organization providing that data wishes to associate with it. Another element of our approach is articulating the context in which the query is made. The context of the query minimally includes who is asking for the information, and for what purpose. More generally, it includes an identification of the person or entity which initiated the query, their role in some (predefined) hierarchy which the data store understands, the group(s) to which they belong, and the intended use of the information. In this sense, we capture the concepts associated with usage [6] and group based controls [7]. In order to address privacy concerns, organizations that collect personal data during their routine business prepare and publish privacy policies to assure their clients. These privacy policies determine the way, modalities, quantum, time period after which, conditions/situation under which, and with whom such personal information can be shared. We note that these policies are generally not machine interpretable or formal policies. However, by making them machine interpretable, we can reason over these policies, and the query context, to decide if the data can be shared. An important feature of the approach is the system of automatic periodic audit to check whether the privacy policies were correctly enforced or not, and identify cases of exception. This is particularly useful in cases where information is shared with 'after-access' obligations, for instance those that maintain that the data would only be used for the stated purpose. The audit component helps to assure the database owners that their privacy policies are being complied with by the user who queried for the data.

A similar approach is used to handle the case of using services on the web or the cloud to store data and perform computations (such as analytics) on it. The claim is that by removing complexity and management issues from the user end, a lower total cost of ownership and greater efficiencies can be realized by cloud based services. Many organizations however face a major barrier to adopting such systems -- they have complex internal policies, as well as legal and statutory constraints on how they handle their data that must be enforced. Such policies are today enforced on internal resources (like data centers) controlled by the organization. For instance, a policy might say that the data must be stored under a certain jurisdiction. When acquiring remote cloud services, it today requires significant human intervention and negotiation -- people have to check whether a provider's service attributes ensure compliance with their organization's constraints. This can get very complex if the provider is composing services using components provided by third parties distributed across the web.

Another concern that organizations have for cloud based services is with security and privacy of the data on the cloud. Since most of the cloud based services allow multiple users at the same time (multi-tenancy), organizations are reluctant to use cloud services for their business critical applications. A semantically rich policy-based framework that manages the cloud data access and security permissions can help elevate these concerns.

Our approach includes a methodology to address the lifecycle issue for virtualized services delivered from the web or the cloud [30], including elements related to data management. This lifecycle provides ontologies [31] to describe data, services and their attributes. In particular, we use semantically rich descriptions of the requirements, constraints, and capabilities that are needed at each phase of the lifecycle [29]. Policies can be described using the same ontology terms so that compliance checks can be automated. This methodology is complementary to previous work on ontologies, e.g., OWL-S, for service descriptions in that it is focused on automating processes needed to procure services on the cloud.

We realize the overall model using OWL (Web Ontology Language) [8] as our semantic description language for the data and query context using ontologies that we have developed [28]. We use Jena [9] as our reasoning infrastructure, and Jena Rules are used to describe policies.

We have developed and implemented a cloud storage service prototype to demonstrate and evaluate our methodology. We used Semantic Web technologies such as OWL, RDF, and SPARQL to develop this tool. The prototype allows cloud consumers to discover and acquire disk storage on the cloud by specifying the service constraints, security policies and compliance policies via a simple user interface. This prototype was developed as part of our collaboration with National Institute of Standards and Technology (NIST).

IMPLEMENTATION

We use a smart cloud broker based approach to address the problem of encouraging the use of web/cloud services. When acquiring web or cloud based services, the consumer organization identifies the technical and functional specifications that a service needs to fulfill. In addition, they specify the organizational policies and legal constraints relating to data usage and management, and security/privacy policies for the service. Service compliance policies such as required certifications, standards to be adhered to, etc. are also identified. Depending on the service cost and availability, a consumer may be amenable to compromise on the service quality. Once the consumers have identified and classified their service needs, they issue a Request for Service (RFS) to a cloud broker service. This RFS uses the ontologies we have developed [30,31] to specify elements of the service acquisition process, as well as security and usage constraints.

The broker engine queries various service providers to match the service domain, data type, compliance needs, functional, and technical specifications; and returns the result with the service providers in priority order. If a consumer finds the exact service meeting their constraints, they can begin consuming the service. Otherwise, the consumer and the service provider will have to negotiate on the service constraints and policies to be met. Service acceptance is usually guided by the Service Level Agreements (SLA) that the service provider and consumer agree upon. A side effect of the negotiation process is that machine understandable SLAs specified in our ontology are automatically generated [32], and can be used for monitoring compliance.

At times, the service provider will need to combine a set of services or compose a service from various components delivered by distinct service providers in order to meet the consumer's requirements. Hence, service negotiation also includes the discussions that the main service provider has with the other component providers. When the services are provided by multiple providers (composite service), the primary provider interfacing with the consumer is responsible for composition of the service.

For the information gathering aspect of the data usage management problem, a compliance checker, similar in concept to the broker above, is used. In our prototypes we have focused on a centralized entity. In ongoing efforts, we are investigating methods to distribute this component. Our ontology describes the notion of hierarchical position level, group, and use. We have adopted description logics (DL), specifically OWL, and associated inferring mechanisms to develop the model and policies. The requester information consists of his position in the hierarchy, his group membership and use for which information is being sought. In our system this information is represented in N3 [15] using the NAT ontology we have developed. The *Nat* ontology defines various properties such as '*belongs_to_hierarchyLevel', 'has_designation'* and *'belongs_to_group'* that can be used to represent the requester details. FOAF [25] is used to allow individuals to describe personal information about themselves and their relationships. This information is used to determine whether the requester has the permission to access the query result based on data owner's (or provider's) privacy policies. The reasoning engine performs reasoning over this information and privacy policies. Our system uses the Jena Semantic Web framework [26] [27] for reasoning over the context data and the policies. These reasoners are used to infer additional facts from the existing knowledge base coupled with ontology and rules. The instance of such reasoner with a ruleset can be bound to a data model and used to answer queries about the resulting inference model. In our system, the reasoning engine uses the *Nat* ontology and the FOAF ontology to represent the requester information, and privacy policies represented in the Jena rule language to generate an inference model. This inference model is used to decide whether the information can be released to requester.

CONCLUSION

The model described above addresses the usage management and control concerns in a multi-user and multi-database owner environment. It addresses both the data gathering issues (where information is gathered from multiple sites and combined to make inferences) and the cloud/web service issue (where data has to be shared with a service provider on the web).

REFERENCES

[1] U.S. General Accounting Office, "Data Mining: Federal Efforts Cover a Wide Range of Uses", (GAO-04-548), May 2004, at 3, 27-64, http://www.gao.gov/new.items/d04548.pdf.

[2] Department of Homeland Security, "Report to Congress on the Impact of Data Mining Technologies on Privacy and Civil Liberties", 7 (2007), http://www.dhs.gov/xlibrary/assets/privacy/privacy_rpt_datamining_2007.pdf.

[3] Department of Homeland Security, "Report to Congress on the Impact of Data Mining Technologies on Privacy and Civil Liberties" 8 (2006), http://www.dhs.gov/xlibrary/assets/privacy/privacy_data_%20mining_%20report.pdf;

[4] Conference Report Cantigny Conference Series, "Counterterrorism Technology and Privacy", McCormick Tribune Foundation, 2005, http://www.mccormickfoundation.org/publications/counterterrorism.pdf

[5] Gio Wiederhold, "Mediators in the Architecture of Future Information Systems", IEEE Computer, March 1992, pages 38-49.

[6] Jaehong Park, Ravi Sandhu "The UCON$_{ABC}$ usage control model", ACM Transactions on Information and System Security (TISSEC) Volume 7 Issue 1, February 2004ACM New York, NY, USA

[7] R. Krishnan, R. Sandhu, J. Niu, and W. H. Winsborough. "A conceptual framework for group-centric secure information sharing". In ASIACCS '09: Proceedings of the 4th International Symposium on Information,Computer, and Communications Security, pages 384–387, New York, NY, USA, 2009. ACM.

[8] W3C, "OWL Web Ontology Language", February 2004, http://www.w3.org/TR/owl-features/

[9] "VOID – Vocabulary of Interlinked Datasets", http://semanticweb.org/wiki/VoiD

[10] Daniel J. Weitzner, Harold Abelson, Tim Berners-Lee, Chris Hanson, James Hendler, Lalana Kagal, Deborah L. McGuinness, Gerald Jay Sussman, K. Krasnow Waterman, "Transparent Accountable Data Mining: New Strategies for Privacy Protection", MIT CSAIL Technical Report-2006-007, http://www.w3.org/2006/01/tami-privacy-strategies-aaai.pdf

[11] Lalana Kagal, Chris Hanson, Daniel Weitzner, "Using Dependency Tracking to Provide Explanations for Policy Management", IEEE Policy: Workshop on Policies for Distributed Systems and Networks, June 2008

[12] Tim Moses. eXtensible Access Control Markup Language TC v2.0 (XACML), February 2005.

[13] Sushil Jajodia, Pierangela Samarati, V. S. Subrahmanian, and Elisa Bertino. "A unified framework for enforcing multiple access control policies" In Proceedings of ACM SIGMOD International Conference on Management of Data, pages 474–485. ACM Press, 1997.

[14] Lalna Kagal and Tim Berners-lee. "Rein : Where policies meet rules in the semantic web", Technical report, Laboratory, Massachusetts Institute of Technology, 2005.

[15] TimBerners-Lee and Dan Connolly, "Notation3 (N3): A readable RDF syntax", Technical report, 2008.

[16] Tim Berners-Lee, Dan Connolly, Eric Prud'hommeaux, and Yosi Scharf, "Experience with n3 rules", In Rule Languages for Interoperability, 2005.

[17] Tim Berners-Lee, "Cwm - a general purpose data processor for the semantic web".

[18] Lalana Kagal, Chris Hanson, and Daniel Weitzner, "Using dependency tracking to provide explanations for policy management", In Proc. IEEE Workshop on Policies for Distributed Systems and Networks, pages 54–61, Washington, DC, 2008. IEEE Computer Society.

[19] Jon Doyle, "Truth maintenance systems for problem solving", Technical report, Cambridge, MA, USA, 1978.

[20] D. Beckett, "Turtle - Terse RDF Triple Language", Technical report, 2007.

[21] D. A. Waterman and F. Hayes-Roth, editors, "Pattern-Directed Inference Systems", 1978.

[22] Gabillon, A. Letouzey, L. Univ. de la Polynesie Francaise, Faaa, French Polynesia, "A View Based Access Control Model for SPARQL",

Network and System Security (NSS), 2010 4th International Conference, September 2010, Melbourne.

[23] Lalana Kagal_ and Joe Pato, “Preserving Privacy Based on Semantic Policy Tools”, IEEE Security & Privacy Magazine Special Issue on: “Privacy-Preserving Sharing of Sensitive Information” August 2010, http://dig.csail.mit.edu/2010/Papers/IEEE-SP/db-privacy.pdf

[24] Mathew Cherian, “A Semantic Data Federation Engine”, MIT Masters Thesis Jan 2011

[25] “The Friend Of A Friend (FOAF) Project “,http://www.foaf-project.org/

[26] “Jena – Semantic Web Framework for Java”, http://jena.sourceforge.net/

[27] Carroll et al, “Jena: implementing the semantic web recommendations”, ACM, pages 74-83, 2004

[28] Madan Oberoi, Pramod Jagtap, Anupam Joshi, Tim Finin and Lalana Kagal, “Information Integration and Analysis: A Semantic Approach to Privacy”, Proc. Third IEEE International Conference on Information Privacy, Security, Risk and Trust (PASSAT), Boston, MA, Oct 2011

[29] Karuna Joshi, “Automation of Service Lifecycle on the Cloud by Using Semantic Technologies”, Proceedings of tenth International Semantic Web Conference, Part II, pp 285-292, Bonn, Oct 2011

[30] K. Joshi , T. Finin , Y. Yesha, "Integrated Lifecycle of IT Services in a Cloud Environment", in Proceedings of The Third International Conference on the Virtual Computing Initiative (ICVCI 2009), Research Triangle Park, NC, October 2009

[31] K. Joshi, OWL Ontology for Lifecycle of IT Services on the Cloud, 2010, http://ebiquity.umbc.edu/ontologies/itso/1.0/itso.owl

[32] US Federal Cloud Computing Initiative, http://www.info.apps.gov/node/2, retrieved on Feb 28 2012

An Information Accountability Framework for Shared eHealth Policies

Randike Gajanayake
Queensland University of Technology
Brisbane, Australia
g.gajanayake@qut.edu.au

Renato Iannella
NEHTA
Brisbane, Australia
renato.iannella@nehta.gov.au

Tony Sahama
Queensland University of Technology
Brisbane, Australia
t.sahama@qut.edu.au

ABSTRACT

Privacy issues have hindered the evolution of e-health since its emergence. Patients demand better solutions for the protection of private information. Health professionals demand open access to patient health records. Existing e-health systems find it difficult to fulfill these competing requirements. In this paper, we present an information accountability framework (IAF) for e-health systems. The IAF is intended to address privacy issues and their competing concerns related to e-health. Capabilities of the IAF adhere to information accountability principles and e-health requirements. Policy representation and policy reasoning are key capabilities introduced in the IAF. We investigate how these capabilities are feasible using Semantic Web technologies. We discuss with the use of a case scenario, how we can represent the different types of policies in the IAF using the Open Digital Rights Language (ODRL).

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection - *access control*; D.3.m [**Programming Languages**]: Miscellaneous; K.4.1 [**Public Policy Issues**]: Privacy; K.5.1 [**LEGAL ASPECTS OF COMPUTING**]: Hardware/Software Protection - p*roprietary rights.*

General Terms

Management, Design, Security, Human Factors, Standardization, Languages

Keywords

E-health, Semantic Web, ODRL, Privacy, Information Accountability

1. INTRODUCTION

E-health is the use of Information and communications technology (ICT) in healthcare. Amongst others, the Internet is the primary mode of communication for e-health applications. The Web is gradually transforming to what is called "the Semantic Web" where the traditional Syntactic Web is leveraged towards a distributed knowledge repository. The semantic web is based on the Resource Description Framework (RDF) [1] for metadata semantics and the Web Ontology Language (OWL) [2] for web ontologies. These technologies enable the development of Web based information systems that are capable of automated reasoning, impossible with the syntactic web. These capabilities open new avenues for e-health systems. But, with the use of the Internet to manage health information, the existing concerns in healthcare such as information security and informational privacy become paramount issues needing rigorous attention. This raise questions as to what the relevant security measures are and how an assurance of privacy can be given to the stakeholders (patients and healthcare professionals). In this paper we present an information accountability framework (IAF) for e-health systems. This framework will make applications such as the one proposed by Gajanayake et al. [3] practicable. We consider requirements of different stakeholders in healthcare and accordingly construct our IAF adhering to information accountability principles in the healthcare context.

The rest of this paper is organised as follows. In the next section we will discuss privacy and its impact on e-health. In section 3 we give a brief account on information accountability and the principles behind the concept. In section 5, we present an IAF for e-health systems by extending an access control model from recent work which is summarised in section 4. Section 6 discusses how the introduced capabilities are attainable with available technologies. We will use a simple case scenario to operationalise the concept.

2. E-HEALTH AND PRIVACY

An eHR is a complete record of a patient's medical history. They may also include information pertaining to sensitive concerns such as sexual health, mental health, addictions to drugs or alcohol, abortions etc. Hence unlawful disclosure of personal information could cause the subject of the information embarrassment and may affect insurability, child custody cases, and even employment [4, 5]. Therefore, informational privacy is vital to ensure the reliability of eHR systems. As a result patients demand strong security for their eHRs. Definitions for privacy come in many different forms. Alan Westin, in his book "*Privacy and Freedom*", defines privacy as "*the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others*" [6], i.e. control of private information. A considerable degree of control over ones personal information is an essential aspect to protecting information privacy [7]. Due to the disparity of data ownership in healthcare, giving control of the data must be handled with care.

Various methods have been proposed to address the privacy conundrum ranging from strict access control to privacy-

DUMW2012, April 16 2012, Lyon, France.

preserving algorithms. Access control mechanisms either permit or deny access, there are no intermediate states. They are not policy-aware and may also hinder the actions of legitimate users of an information system [8]. According to Kagal et al. [9] relying solely on access control mechanisms to guard information would be inadequate for privacy protection.

Information accountability (IA) can complement access control mechanisms and support policy-awareness. The principles behind IA, in theory, would make sure that information users follow the appropriate rules and policies. To facilitate IA principles, systems should implement usage policies on its assets. Considering data in eHRs digital assets digital rights management (DRM) techniques can be used for the management of the data. Privacy policies in e-health can be represented using an appropriate digital rights expression language (REL). Policies on the use of data in an eHR can be set by the patient, a trusted healthcare representative, a health authority or all the above.

3. INFORMATION ACCOUNTABILITY

Accountability systems lack formal foundations making it an attractive theme to many [10-13]. Jagadeesan et al. [10] assume that the relevant privacy policies exist and develop formal foundations for information accountability in terms of the privacy policies which define appropriate sharing of information among agents and provide algorithms that can be used by an auditor to check for compliance with rules. Weitzner et al. [14] propose a solution to the question of compliance of privacy policies by tracking all transactions and making them transparent hence creating an incentive for the users to abide by the rules. They assume that appropriate policy rules exist with a formal representation, policy-aware transaction logs and a policy-reasoning capability which would enable accountability systems to hold information users accountable for misuse. Focusing on the facts Weitzner et al. [14] put forth, Sloan et al. [13] address information accountability in terms of both social policies and technical aspects. They point out difficulties to developing accountability systems by stating that automated checking for compliance of privacy policy is a necessity for accountability systems and without the adequate foundations in both formal models and public policy issues they are unlikely to do so. They believe that policies required to developing accountability systems are informational norms and state that a proper balance between privacy requirements and competing concerns is necessary to sustain the architectural and social aspects introduced by Weitzner et al. [14].

Access control and accountability are closely related concepts. Access control is about restrictions, whereas accountability is about punishment. Hence for accountability systems, *audit logs* are essential [15]. Accountability systems facilitate *fair use* of information. Rather than prevention via rigid locks on data, accountability is about *deterrence*. The presence of an accountability mechanism delivers a threat of punishment which would deter users from intentional misuse. Accountability systems should facilitate *transparency* such that all relevant parties have the capability to observe how information is used and by whom. This makes *bad acts visible* and helps deter users from misuse [14]. The users of an accountability system should be *well informed*, i.e. a notification process where users are informed about underlying policies before an action occurs should be is in place. For example a user will be notified whether he is actually authorised to use a particular set of data he is trying to use and the ramifications if he proceeds regardless of the usage policies in place. This will also help in facilitating non-repudiation which is a significant aspect in information security. When holding someone accountable, the trustworthiness of the data about the inappropriate transaction is critical. Hence, *provenance* of data and metadata is a significant factor in information accountability. Electronic data does not have the necessary historical information that would help end-users, reviewers or regulators make the necessary verifications [16]. In an accountability system provenance can be facilitated using appropriate transaction logs. These transaction logs also serve another purpose in terms of accountability by being *policy-aware*. Policy-aware transaction logs can also facilitate *policy reasoning* capabilities and enable the users to reason about misuse and against claims of misuse.

Creating proper *incentives* that would make consumers follow rules of accountability systems is important [13]. For an information user, the threat of punishment is an incentive to follow system rules. An incentive such as a strong assurance of privacy should be given to patients to prevent them from withholding information or enforcing rigid restrictions on data.

3.1 Information accountability in healthcare

In order to understand the concept of information accountability in healthcare, it is important to clearly identify the different parties in healthcare that can be held accountable, the issues for which a party can be held accountable and the appropriate mechanisms for accountability in healthcare [17]. Policies should be developed that address the different capabilities of roles within the industry. These policies should capture the requirements of all relevant parties. As stated above, in the healthcare domain it is difficult to define who owns health information. It is clear that patients are the subjects of health information. Patients are not always medical professionals; hence it is impossible to give them full control of their health information. Privacy policies should accompany an input from a professional health body such as a trusted medical practitioner or a central health authority. But is it important to balance between the patient's privacy requirements and the requirements of the healthcare providers or the care givers (competing concerns). In a healthcare setting the patients privacy policies cannot contradict those set by the healthcare providers or the health authority. The IMIA code of ethics for medical information professionals [18] states under their first ethics principle; *Principle of Information-Privacy and Disposition* that "*all persons have a fundamental right to privacy, and hence to control over the collection, storage, access, use, communication, manipulation and disposition of data about themselves"*. A patient with an eHR, hence, should have the following capabilities; 1) the capability to allow a selected group of medical professionals to access the eHR, 2) the capability to hide certain health information from particular health practitioners who already have access to their eHR, 3) the capability to view and how the data in the eHR is used by authorised personnel, 4) the capability to inquire about potential misuse of data. The data consumers (health professionals and health authority) also have particular requirements. We can identify them as follows; 1) the capability to define security policies within the organization, 2) access to the relevant information in a non-restrictive and timely manner, 3) the capability to share patient health information with other health specialists, 4) the capability to override patients' security settings in special circumstances (e.g. life threatening emergency situations, mental health related situations). It is important to note that usage policy enforcement might not always be beneficial to the patient. While fulfilling these privacy requirements under no circumstance must the health of the patient is compromised. A clear procedure for overriding usage policies in emergency situations should be defined. The nature of the healthcare domain may forces the implementation of a *break the glass* approach in emergency situations. The policy formulation process must consider the requirements of both parties. A compromise between these requirements must entail the final policy representation of the systems and the proper integration of these policies would improve patient confidence in the system.

Apart from the requirements stated above, certain circumstances might requirement some health conditions be kept hidden from the patients. For example this may be the case for patients suffering from severe mental health conditions where the knowledge of particular illnesses may aggravate existing health conditions. They may also be considered unfit to manage their eHR. We acknowledge this eventuality but consider them as rare occurrences and do not integrate such capabilities in to the framework. However, in such cases the control over the patient's eHR may be given to a custodian or a trusted health professional (HP) such as the patients GP who can take the patient's role in controlling the eHR.

4. PRIVACY ORIENTED ACCESS CONTROL FOR EHR

Following is a brief description of the access control model in [19]. The access control model uses a combination of the principles behind MAC, DAC, RBAC and PBAC. This was done in order to fulfill the requirements of different stakeholders discussed above. The basic protocol for the proposed access control system is illustrated in Figure 1. We assume that the patient has a comprehensive eHR under a relevant health authority.

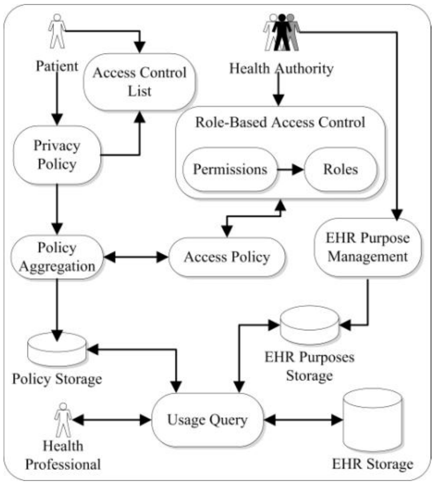


Figure 1. Privacy oriented access control model architecture

The eHR is formulated such that each type of data in the eHR (e.g. identity data, general health data, dental health data, mental health data, etc.) can be distinguished by eHR data type identifiers. For each of these data types there exists a set of predefined purposes for which the data can be used that are defined by a central health authority. The patient and the health authority can set privacy and access policies respectively. These two policies are later combined using the *Policy Aggregation* to form the final operational policy. The protocol for the policy formulation is as follows.

The health authority defines intended purposes and sensitivity labels for each data type and element. We use object sensitivity labeling using a tree structure (a sensitivity tree (ST)) that has the eHR itself as the root element, the data types as children and data elements as grandchildren. A sensitivity label is not assigned to the objects themselves rather we relate the access level of a particular user (health professional) in terms of the sensitivity label of the data elements. Note that the sensitivity labels mentioned here are different from the classical hierarchical security levels found in MAC [20]. The nature of health information makes it difficult to define a clear hierarchical structure for the sensitivity of data elements that is general to all patients. For example, sexual health and mental health information may have the same sensitivity for some patients and may not be so for others.

Definition: A sensitivity label (SL) is a tuple <ASL, PSL>, where ASL = $\{asl_1, asl_2 \ldots asl_n\}$ is a set of allowed sensitivity labels and PSL = $\{psl_1, psl_2 \ldots psl_n\}$ is a set of prohibited sensitivity labels.

ASL = $\{asl_i\}$; i = 1…n is denoted as all of the descendants of asl_i including asl_i.

PSL = $\{psl_j\}$; j = 1…n is denoted as all of the descendants of psl_j including psl_j.

Example: Matt can access Gary's mental health details but cannot access his Sexual or Dermatology details. The access level for Matt can be represented in terms of sensitivity labels as follows.

SL_{Matt} = < {eHR}, {Sexual Health, Dermatology Health} >

Here we use the Denial-Takes-Precedence principle. Access is granted to the entire eHR and then access is denied to specific field by the PSL. This helps isolate the most sensitive information in the eHR that need to be hidden from certain users.

Patients and the health authority set sensitivity levels for health professionals. The sensitivity level defined by the health authority is different to the ones defines by the patients. PSLs set by the health authority will always be *NULL*. This is because the health authority is concerned with allowing access to data elements. The prohibitions are defined by the patients. The ASL set by the patients always precedes that which is set by the health authority. The ASL set by the health authority always precedes PSL set by the patients if there is a conflict. This feature will ensure that the relevant information is always available to the relevant health professional.

5. INFORMATION ACCOUNTABILITY FRAMEWORK FOR E-HEALTH

Here we present an information accountability framework (*IAF*) for e-health systems. It can be considered as an extension to the access control model described above. In the IAF the policies defined in the access control model act as the underlying policies to which the users must comply to but do not prevent users from accessing data. This is to facilitate unrestricted access to health information for authorised users. The reasoning capability of the IAF takes these policies in to consideration whist performing such tasks.

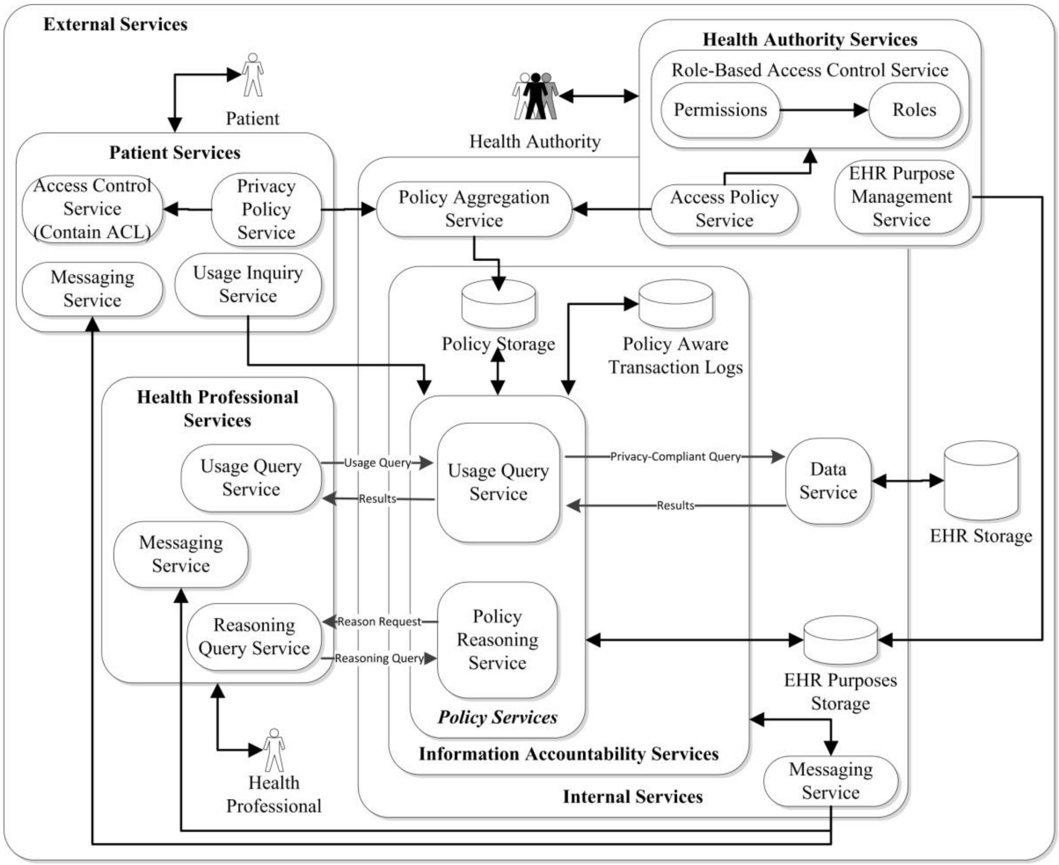


Figure 2: Schematic IAF architecture

The IAF is divided in to two categories of services; external services and internal services and have three types of users; patients (*P*), a health authority (*HA*) and health professionals (*HP*). A schematic architecture is shown in the Figure 2.

Internal services consist of a *policy aggregation service*, the *information accountability services*, a *messaging service*, a *data service*, *policy storage* and the EHR *Purpose storage*. External services of the IAF include *patient services*, *health authority services*, *health professional services* and the external *EHR storage*. Detailed descriptions of these services are given next.

5.1 Internal services

Information accountability services consists of *policy storage* (PS_{IAS}), *policy aware transaction logs* ($PATL_{IAS}$) and policy services containing a *usage query service* (UQS_{IAS}) and a *policy reasoning service* (PRS_{IAS}). PS_{IAS} stores the policies it receives from the *policy aggregator service*. UQS_{IAS} processes the usage queries it receives from health professional services requesting access to EHR data. Once the policy service receives an *inquiry query* from patient services PRS_{IAS} send a request to the health professional service requesting a *reasoning query* for a particular information usage instance. The reasoning queries are processed with the use of $PATL_{IAS}$ which contains all past transactions of the system.

Other internal services include a *policy aggregator service* (PAS_{IS}) which amalgamates the policies from PPS_P and APS_{HA} in such a way that the patient's privacy requirements are met and the health authorities' policies be satisfied, a *data service* (DS_{IS}) which is the only component with access to the EHR storage, a *messaging service* (MS_{IS}) which rends out the relevant messages to other services and an *EHR purposes storage* (EPS_{IS}) which consists of the intended purposed of each of the data types in the *EHR*. The EPS_{IS} is managed by *HA*.

5.2 External services

External services are used by the end users to give inputs to the internal services and receive results from them. External services consist of patient services, health authority services, health professional services and the EHR storage.

Patient services are used by a patient to manage their EHR. The patient services consist of an *access control service (ACS_P)*, *privacy policy service (PPS_P)*, *messaging service (MS_P)* and a *usage inquiry service (UIS_P)*. A patient maintains an access control list (ACL) with the use of ACS_P. The patients set their privacy policies using *PPS* and assign sensitivity levels for trusted health professionals in the ACL. These policies are then amalgamated by the *policy aggregation service* (PAS_{IS}) with the policies of the health authority and stored in PS_{IAF}. Patients receive notifications and can send messages to HPs through the MS_P from the internal services. Notifications include regular updates on the EHR, notifications of information access by HPs, warnings of potential information misuse and messages from HPs.

All messages need to go through the internal services for them to be recorded in the Transaction logs.

Health authority services are used by a central health authority to manage access settings for health professionals. Health authority services consists of a *role based access control service* ($RBACS_{HA}$), an *EHR purpose management service* ($EPMS_{HA}$) and *access policy service* (APS_{HA}). The HA set minimum access levels for HPs using APS_{HA} together with $RBACS_{HA}$. These policies are combined with the patient's privacy policies according to the access control protocol in [19], which is also summarised in section 3. HA uses $EPMS_{HA}$ to manage the EHR purposes in EPS_{IS}.

Health professional services are used by health professionals to access patient EHR information. HPs are able to perform actions such as read and write. HPs are also able to initiate information sharing requests in order to share patient health information with other HPs to make informed decisions. Health professional services include a *usage query service* (UQS_{HP}), a *reasoning query service* (RQS_{HP}) and a *messaging service* (MS_{HP}). HPs can lodge usage queries using UQS_{HP} requesting access to EHR information. These queries contain purposes for which information is required. The queries are processed by the UQS_{IAS} and if they are policy compliant access is granted. If the usage queries are not policy compliant a warning notification is sent to the requester at which point he can either comply with the warning or disregard it. If the warning is disregarded and the data is accessed by the HP, a message is sent by the $\mathrm{MS}_{\mathrm{IS}}$ to MS_{P} notifying the patient of potential information misuse. At this point the patient may initiate a usage inquiry using UIS_{P}. As a result PRS_{IAS} sends a request to RQS_{HP}. The HP then has to send a justification of the use of information in the form of a reasoning query through the RQS_{HP}. The justification is processed by the PRS_{IAS}. If the provided justification is valid the incident is resolved. If not, further action (such as legal action) would be taken which we would not discuss in this paper (a justifiable action would be in the case of an emergency where the existing policies had to be overridden for the sake of the patient's health). PRS_{IAS} should have the capability to deduce whether a provided justification is valid. This process of inquiries and resulting justifications enables the system to detect intentional misuse of data by users.

6. FEASIBILITY

The main capabilities of the IAF are policy representation, policy storage and policy reasoning capabilities. The key challenge in implementing the IAF in a technical point of view is to fulfill these capabilities. In this section we will give an account as to how these capabilities are feasible through available semantic web technologies.

As discussed in section 3, proper representation of policies is vital in information accountability. For our model we look to digital rights management (DRM) as a solution. Apart from their applications in copyright protection of media files, etc on the Internet, DRM technologies are becoming a prominent resource in protecting private information of individuals [21]. DRM has many similarities to the traditional access control model but differs in that they require information to remain protected even after access is granted to authorised users. DRM deals with usage control of a piece of information resource by authorised users. Each piece of information is protected by a usage license created by the digital rights holder. DRM can benefit e-health technologies by providing a means to manage the use of eHRs. Rights expression languages (REL) are a critical aspect of DRM systems. The Open Digital Rights Language (ODRL) version 2 [22] is based on XML and provides a syntax and semantics to express policies related to digital assets. We have chosen ODRL as the policy language for our model because it is independent of implementation constraints and it's capable of expressing a wide range of policy-based information.

6.1 Healthcare scenario

Consider the following scenario. Gary has a comprehensive eHR. Gary has a list of trusted healthcare providers (health professionals) to whom he gives access to data in his eHR. Peter is Gary's GP, Sandra is a dermatologist, Bill is a sexual health specialist and Matt is a mental health specialist who has treated Gary in the recent past. Gary can set privacy settings to govern the access to his eHR. A central health authority can also set access settings to patient's eHR by considering the roles of each health professional. In addition to privacy and access policies, other constraints can be present in the eHR. One such policy can be a *take control* policy. In which an eHR holder may take control of their eHR at the age of 15 (which was previously controlled by a parent, legal guardian of authorised representative) and must take control at the age of 18. Such policies must accompany privacy and access policies in the eHR.

6.1.1 Scenario

After noticing a skin rash, Gary visits his trusted dermatologist Sandra for a check up. The preliminary examination reveals that Gary's skin condition could be linked to a known sexually transmitted disease (STD). Gary does not have a sexual health specialist in his list of trusted health professionals. However, Sandra wants to share Gary's details with a sexual health specialist, Bill, in order to get a specialists opinion on the situation. Bill has a default access level set by the health authority to be able to access patients' sexual health details and dermatology details. Since Sandra is in Gary's list of trusted HPs to be able to access Gary's dermatology information, she can initiate a request to share Gary's details with other health professionals. Gary, however, is notified of this action by Sandra. After Bill gets this request, he initiates a usage request to use the data for diagnosis purposes. At some point during or after this episode of care, Gary may include Bill to his list of trusted health professional.

6.1.2 ODRL policies

Gary allows Sandra to access his EHR but restricts her from accessing his sexual health details and mental health details. Below is an ODRL V2 XML instance of this policy.

```
<o:policy xmlns:o= "http://w3.org/ns/odrl/2"
xmlns:eh="urn:ehealth.gov" type="
http://w3.org/ns/odrl/2/privacy" uid="policy-use-ehr"
conflict="o:prohibit">
  <o:permission>
    <o:asset uid="urn:ehr:gary" relation="o:target"/>
    <o:party uid="urn:patient:gary" role="o:assigner"/>
    <o:party uid="urn:healthPro:dermatHealth:sandra"
      role="o:assignee"/>
    <o:action name="o:read"/>
    <o:constraint name="o:purpose" operator="o:eq"
      rightOperand="eh:healthCare"/>
  </o:permission>
```

```
 <o:prohibition>
   <o:asset uid="urn:ehr:gary" relation="o:target"/>
   <o:party uid="urn:patient:gary" role="o:assigner"/>
   <o:party uid="urn:healthPro:dermatHealth:sandra"
     role="o:assignee"/>
   <o:action name="o:read"/>
   <o:constraint name="o:purpose" operator="o:eq"
     rightOperand="eh:sexualHealthCare"/>
   <o:constraint name="o:purpose" operator="o:eq"
     rightOperand="eh:mentalHealthCare"/>
 </o:prohibition>
</o:policy>
```

The conflict attribute of the policy above is set to "*prohibit*" indicating that prohibitions take precedence in the policy. The health authority can set an access policy for Sandra which is given below.

```
<o:policy xmlns:o= " http://w3.org/ns/odrl/2"
xmlns:eh="urn:ehealth.gov" type="
http://w3.org/ns/odrl/2/agreement" uid="policy-use-ehr">
 <o:permission>
   <o:asset uid="urn:ehr:gary" relation="o:target"/>
   <o:party uid="urn:health:authority" role="o:assigner"/>
   <o:party uid="urn:healthPro:dermatHealth:sandra"
     role="o:assignee"/>
   <o:action name="o:read"/>
   <o:constraint name="o:purpose" operator="o:eq"
     rightOperand="eh:dermatHealthCare">
   <o:constraint name="o:purpose" operator="o:eq"
     rightOperand="eh:sexualHealthCare">
 </o:permission>
</o:policy>
```

The health authority is responsible for setting default access policies for healthcare roles, in this case for the role of a dermatologist. In the policy above HA gives Sandra the permission to access Gary's dermatology details and sexual health details. Note here that Gary's settings prohibit Sandra from accessing his sexual health details. But we assume a hypothetical scenario where a relationship between skin conditions and STDs exist, and every dermatologist should have access to the patient's sexual health details. The health authority is aware of this fact and allows all dermatologists access to patients sexual health details. According to the access control protocol in section 3, the settings by the health authority always prevail over patient settings. The final policy will be a combination of the two policies and hence the requirement for PAS_{IS} in the IAF. The amalgamated policy for Sandra is given below.

```
<o:policy xmlns:o= " http://w3.org/ns/odrl/2"
xmlns:eh="urn:ehealth.gov" type="
http://w3.org/ns/odrl/2/privacy" uid="policy-use-ehr"
conflict="o:prohibit">
 <o:permission>
   <o:asset uid="urn:ehr:gary" relation="o:target"/>
   <o:party uid="urn:patient:gary" role="o:assigner"/>
   <o:party uid="urn:healthPro:dermatHealth:sandra"
     role="o:assignee"/>
   <o:action name="o:read"/>
   <o:constraint name="o:purpose" operator="o:eq"
     rightOperand="eh:healthCare"/>
 </o:permission>
 <o:prohibition>
   <o:asset uid="urn:ehr:gary" relation="o:target"/>
   <o:party uid="urn:patient:gary" role="o:assigner"/>
   <o:party uid="urn:healthPro:dermatHealth:sandra"
     role="o:assignee"/>
   <o:action name="o:read"/>
   <o:constraint name="o:purpose" operator="o:eq"
     rightOperand="eh:mentalHealthCare"/>
 </o:prohibition>
</o:policy>
```

This final policy is stored in PS_{IAS} and is used by other services. Updates are done to the policies in PS_{IAS} accordingly.

Information sharing is an important aspect of healthcare and is facilitated in the IAF. HPs who are already in the ACL can initiate sharing requests.

```
<o:policy xmlns:o="http://odrl.net/2.0"
xmlns:eh="urn:ehealth.gov" type="
http://w3.org/ns/odrl/2/request" uid="policy-share-ehr">
  <o:permission>
    <o:asset uid="urn:ehr:gary" relation="o:target"/>
    <o:party uid="urn:healthPro:dermatHealth:sandra"
      role="o:assignee"/>
    <o:action name="o:share"/>
    <o:constraint name="o:purpose" operator="o:eq"
      rightOperand="eh:dermatHealthCare">
    <o:constraint name="o:recipient" operator="o:eq"
      rightOperand="urn:healthPro:dermatHealth:bill">
  </o:permission>
</o:policy>
```

In the policy above Sandra initiates a request to share Gary's dermatology details with Bill. Bill accepts this request by lodging the following access request to read Gary's dermatology details. Requests resulting from sharing requests are allowed (holding to general access policies) since the initial request was from a HP already in the ACL.

```
<o:policy xmlns:o= " http://w3.org/ns/odrl/2"
xmlns:eh="urn:ehealth.gov" type="
http://w3.org/ns/odrl/2/request" uid="policy-use-ehr">
  <o:permission>
    <o:asset uid="urn:ehr:gary" relation="o:target"/>
    <o:party uid="urn:healthPro:sexualHealth:bill"
      role="o:assignee"/>
    <o:action name="o:read"/>
    <o:constraint name="o:purpose" operator="o:eq"
      rightOperand="eh:dermatHealthCare">
  </o:permission>
</o:policy>
```

Using ODRL we can formulate the different types of policies within the eHR system. But the Challenge lies in using ODRL in the Semantic Web domain. Next we will look at how we can use ODRL in conjunction with semantic web technologies and how we can attain the capabilities proposed for the IAF.

6.2 ODRL in the Semantic Web

ODRL is a solution to move DRM to the Internet. But in order to enforce the semantics of the policies in conjunction with ODRL, a corresponding ontology is required. At the time of writing such ontology was not present. The ontology for the policies can be represented using OWL. Even though a comprehensive ontology for ODRL V2 is required for an end , we will not present one in this paper. Such ontologies allow us to achieve the reasoning capabilities proposed in the IAF.

EPS_{IS} contains an ontology representing the relationships between the eHR data themselves and eHR data and the intended purposes.

This ontology together with a comprehensive medical ontology enables us to infer facts otherwise would not be available. For example, the presence of the fact that Gary has a particular allergy in the EPS_{IS} can lead to the inference of the fact that a particular medication has the tendency to be harmful to Gary. This fact would not have been available to the eHR system without a specific external input saying so or if Gary has had an illness which is usually treated by this particular medication. The inferences are updated with new data and facts available to EPS_{IS}. The policies in PS_{IAS} are stored in RDF with vocabularies from the ODRL ontology. The queries made by UIS_P and PRS_{IAS} are made in a RDF query language like SPARQL [23]. Data stored in $PATL_{IAS}$ is also in RDF allowing mining to be done using SPARQL. Together with these services and a policy aware reasoner, PRS_{IAF} allow us (with a suitable natural language translation middleware) to process queries such as "*Why did Sandra read my sexual health details?*" by Gary. Similarly, Sandra will be able to justify why she read Gary's sexual health details. The validity of the justification is determined after mining the $PATL_{IAS}$ and PS_{IAS}. A provided justification holds if the facts confirm with the available knowledge. Note here that as mentioned above, the patient can only lodge an inquiry query if there has been a possible misuse of data i.e. some underlying policy has been violated by the user. The justification is on why the user has done so. The ontologies defined enable us to infer facts that validate the justification. For example, in an emergency situation the treating health professional will access all necessary information from the eHR regardless of the privacy and access policies. This will be recorded in $PATL_{IAS}$. For any inquiry made by the patient to clarify data usage related to this episode of care, the fact that the incident was considered and recorded as an emergency would validate the justifications given by the health professionals.

7. CONCLUSION AND FUTURE WORK

We have presented an information accountability framework (IAF) for e-health which adheres to information accountability (IA) principles and requirements of stakeholders in healthcare. IA is a term better defined contextually rather than in a general sense. We focused on the healthcare context and treated each IA principle accordingly. The requirements of the healthcare domain which we considered are mainly privacy requirements of patients and access and usage requirements of health professionals. Amongst others these carry the most potential to hinder the development of e-health systems and are the main concerns of consumers of those systems. In any accountability system, policy representation is clearly a key aspect. In our model we used ODRL as the policy language and discussed how we can represent the different privacy and access policies in the IAF. Semantic Web based policy management has been studied by many and some attractive solutions have been proposed [24-26]. However, we chose ODRL as the policy language for our model to give us the flexibility needed to extend the existing model to suit the capabilities introduced in our model. For example using an extension to ODRL we can represent policy aware transaction logs which are then used for reasoning to detect misuse.

Policy reasoning is the other key factor in information accountability. Currently the only technologies that provide such capabilities and are readily available are semantic web technologies. We discussed how we can use semantic web technologies such as OWL ontologies and RDF to develop the proposed IAF. It is clear that developing a comprehensive eHR system with an IAF is an immense undertaking. But with the level of technology currently at the disposal of developers it is without a doubt feasible task.

In e-health, accountability systems will enable the use of health information in a more free but controlled manner. This will allow health professionals to access relevant information at any point without the restrictions currently present in e-health solutions. We believe that the presence of the IAF will increase the confidence of the patients towards e-health systems and would lead to e-health systems being better adopted. Barriers still exist in our venture towards building a working system with the capabilities introduced. We are currently working on demonstrating the presented IAF using the technologies discussed. At the time of writing the development of the aforementioned ontologies are ongoing. Building a comprehensive eHR system is not our goal. Our goal is to show that with IA capabilities the current state of e-health systems can be improved to a more open and healthcare oriented state from a security and privacy oriented state.

8. ACKNOWLEDGEMENTS

We would like to thank the National Information and Communications Technology Australia (NICTA) for partially funding this ongoing research project.

9. REFERENCES

[1] Lassila, O. and Swick, R. *Resource Description Framework (RDF) Model and Syntax Specification*. 1999.

[2] McGuinness, D. and van Harmelen, F. *OWL Web Ontology Language Overview*. City, 2004.

[3] Gajanayake, R., Iannella, R. and Sahama, T. Sharing with Care: An Information Accountability Perspective. *Internet Computing, IEEE*, 15, 4 2011), 31-38.

[4] Pratt, W., Unruh, K., Civan, A. and Skeels, M. M. Personal health information management. *Commun. ACM*, 49, 1 2006), 51-55.

[5] Cannoy, S. D. and Salam, A. F. A framework for health care information assurance policy and compliance. *Commun. ACM*, 53, 3 2010), 126-131.

[6] Westin, A. *Privacy and Freedom*. New York Atheneum, 1967.

[7] Solove, D. J. Understanding Privacy. *Daniel J. Solove, UNDERSTANDING PRIVACY, Harvard University Press, May 2008*2008).

[8] Kagal, L. and Pato, J. Preserving Privacy Based on Semantic Policy Tools. *Security & Privacy, IEEE*, 8, 4 2010), 25-30.

[9] Kagal, L. and Abelson, H. *Access Control is an Inadequate Framework for Privacy Protection*. City, 2010.

[10] Jagadeesan, R., Jeffrey, A., Pitcher, C. and Riely, J. *Towards a Theory of Accountability and Audit Computer Security – ESORICS 2009*. Springer Berlin / Heidelberg, City, 2009.

[11] Feigenbaum, J., Hendler, J., Jaggard, A. D., Weitzner, D. J. and Wright, R. N. *Accountability and Deterrence in Online Life*. City, 2011.

[12] Feigenbaum, J., Jaggard, A. D. and Wright, R. Towards a Formal Model of Accountability. In *Proceedings of the New Security Paradigms Workshop* (CA, USA, September 12-15, 2011), [insert City of Publication],[insert 2011 of Publication].

[13] Sloan, R. H. and Warner, R. Developing Foundations for Accountability Systems: Informational Norms and Context-Sensitive Judgments. *Annual Computer Security Applications Conference, Workshop on Governance of Technology, Information, and Policies, 2010*2010).

[14] Weitzner, D. J., Abelson, H., Berners-Lee, T., Feigenbaum, J., Hendler, J. and Sussman, G. J. Information accountability. *Commun. ACM*, 51, 6 2008), 82-87.

[15] Lampson, B. Privacy and security: Usable security: how to get it. *Commun. ACM*, 52, 11 2009), 25-27.

[16] Moreau, L., Groth, P., Miles, S., Vazquez-Salceda, J., Ibbotson, J., Jiang, S., Munroe, S., Rana, O., Schreiber, A., Tan, V. and Varga, L. The provenance of electronic data. *Commun. ACM*, 51, 4 2008), 52-58.

[17] Emanuel, E. J. and Emanuel, L. L. What Is Accountability in Health Care? *Annals of Internal Medicine*, 124, 2 (January 15, 1996 1996), 229-239.

[18] International Medical Informatics Association *IMIA Code of ethics for health information professionals*. City, 2002.

[19] Gajanayake, R., Iannella, R. and Sahama, T. Privacy Oriented Access Control for Electronic Health Records. In *Proceedings of the Workshop on Data Usage Management on the Web* (Lyon, France, 2012).

[20] Sandhu, R. S. and Samarati, P. Access control: principle and practice. *Communications Magazine, IEEE*, 32, 9 1994), 40-48.

[21] Feigenbaum, J., Freedman, M. J., Sander, T. and Shostack, A. Privacy Engineering for Digital Rights Management Systems. *Lecture Notes in Computer Science, Security and Privacy in Digital Rights Management*, 23202002), 76-105

[22] ODRL Initiative *ODRL V2.0 - Core Model - Working Draft*. City, 2012.

[23] Prud'hommeaux, E. and Seaborne, A. *SPARQL Query Language for RDF*. City, 2008.

[24] Kagal, L., Finin, T. and Joshi, A. *A Policy Based Approach to Security for the Semantic Web*. Springer Berlin / Heidelberg, City, 2003.

[25] Kagal, L., Finin, T. and Anupam, J. *A policy language for a pervasive computing environment*. City, 2003.

[26] Tonti, G., Bradshaw, J., Jeffers, R., Montanari, R., Suri, N. and Uszok, A. *Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAoS, Rei, and Ponder*. Springer Berlin / Heidelberg, City, 2003.