Technische Universität München (TUM)

Fakultät für Informatik

Institut für medizinische Statistik und Epidemiologie

Lehrstuhl für medizinische Informatik

Pseudonymization in Biomedical Research

Ronald Rene Lautenschläger, M.Sc.

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

Vorsitzende(r):	Prof. Dr. Florian Matthes
Prüfer der Dissertation:	1. Prof. Dr. Klaus A. Kuhn
	2. Prof. Dr. Claudia Eckert

Die Dissertation wurde am 29.08.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 29.01.2019 angenommen.

Acknowledgement

Firstly, I would like to express my sincere gratitude to my advisor Prof. Klaus A. Kuhn for helping me finding my research issue and also for the continuous support of my Ph.D study, for his patience and his guidance. Besides my first advisor, I would like to thank my second advisor Prof. Claudia Eckert for her insightful comments, encouragement and the feedback provided. My sincere thanks also goes to Dr. Fabian Prasser and Dr. Florian Kohlmayer, from whom my studies have greatly benefited. Without their precious support it would not have been possible to conduct this research. The fruitful discussions, meticulous comments and suggestions of Dr. Prasser, Dr. Kohlmayer and Prof. Kuhn were illuminating and crucial for this work.

I owe my deepest gratitude to my family and my friends who encouraged me during my research and my whole life.

Last but not the least, I would like to thank my co-workers of our institute for their support and friendly assistance in all matters.

Abstract

Background: Biomedical research requires collecting and analyzing sensitive multidimensional datasets. In recent years, the domain has shifted towards collaborative and multidisciplinary approaches. Simultaneously, increasing public awareness of privacy threats has led to high social and political pressure to prevent misuse of personal data. Several national and international data protection laws and regulations demand the separation of data that may identify patients from biomedical data used for research purposes. The process, in which a confidential link between both types of data is maintained, is commonly known as pseudonymization. Besides traditional security and privacy measures, pseudonymization provides additional protection for sensitive personal health information.

Objective: Legal requirements and regulatory recommendations cannot function as a blueprint for implementing pseudonymization. Existing concepts for pseudonymization often do not sufficiently cover details on implementation. As a result, most pseudonymization solutions have been developed empirically, lacking a systematic methodology regarding a risk and threat analysis as well as technical design and implementation. The aim of this work is to provide the groundwork for such a methodology and, in addition, to provide a basis for the comparison of existing solutions. Finally, this work aims at simplifying future developments, by describing core requirements for a reference architecture and implementation options for generic solutions.

Methods: In this thesis, we examine existing pseudonymization concepts and extract basic requirements for managing pseudonymized data and define a reference architecture fulfilling these requirements. In addition, we describe security and privacy requirements for identified or known threats. Secondly, we systematically model the design space of countermeasures for enforcing these requirements.

Results: We identified core requirements and developed a generic solution that supports the collection and management of distributed pseudonymized data. We showed how different combinations of countermeasures result in different pseudonymization architectures with different security and privacy characteristics. Next, we presented a risk and threat analysis for our approach and show which countermeasures were used to guard against identified threats. Our approach has been successfully used to implement a national and an international rare disease network where data about more than 1400 individuals has been collected to date.

Conclusions: This work provides a first step towards a model for pseudonymity in biomedical research. The formal definition of requirements, especially for security and privacy as described in this thesis should become a fundamental basis for designing secure and privacy-preserving research data management systems.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Collaborative data collection and data sharing are crucial to biomedical research. Examples are international projects which aim at making data available to research communities [00a] or to help to create better access to large datasets [PeOm13]. While collaborative research is growing fast a series of publications has shown relevant privacy[1] threats[2] in this context [ACCH13, ApJo10, Mali05]. Well-known regulations which address these aspects are the General Data Protection Regulation (GDPR) of the European Union [Jour16], national regulations for adaptation of GDPR, e.g., [Fede17] and the European Recommendation on Research on Biological Materials of Human Origin [Coun06] as well as in the US the HIPAA Privacy Rule [Usde13]. Data protection and privacy-preserving access to data have become topics of relevance not only for data but also for biomaterial [00b, PKLK15, Publ00, WKWS11].

As longitudinal data collection is required, demographic and nominal data must be stored to allow recognizing individuals during follow-up visits. After the collection of data, the next phase consists of allowing other researchers to get an overview over available data for the purpose of sharing and collaboration. For this *provision of access*, public databases and catalogs providing read-only views with metadata about the available microdata are important resources [PKLK15]. Examples include NCBI's database of Genotypes and Phenotypes (dbGaP) [MFJK07], the BBMRI catalog [WKWS11], BBMRI ERIC [OTBD15] or the Danish National Biobank [Niel12]. When researchers have identified data of interest, they

[1] "the right of […] a person […] to determine the degree to which it [...] is willing to share its personal information with others" [Shir07].

[2] "[a]ny circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service. […]" [Nist06].

request permission for accessing the original dataset. If access is granted, the data is *released* and *shared* with the researcher for further processing under certain restrictions to which a researcher needs to agree. Here, a system must provide read-only access to a representation of the individual-level microdata that is sufficiently detailed to perform the desired analyses. This phase is often supported by the same systems as the previous phase. Finally, the data is integrated, cleaned and *analyzed* [PKLK15].

In all the systems utilized in these phases, special techniques are employed to ensure *confidentiality*[3]*.* The use of protected network communication, strong authentication, role-based access control are popular means to ensure confidentiality by preventing unauthorized access by attackers[4]. Further mechanisms that can be applied on top of the measures mentioned, are measures against *privacy* risks[5]. These measures are de-identification and anonymization. Both methods aim at removing or altering potential identifiers. We will not go into further details regarding anonymization or de-identification for it exceeds the scope of this thesis.

As stated in ISO/TS 25237, anonymization is a measure to irreversibly "remov[e] the association between the identifying data set and the data subject[6]" [Inte00], in order to avoid re-identification of individuals by attackers [PKLK15]. In the remainder of this work, we will use the well-known terminology defined by the *ITU* [Itut10]. According to their *Recommendation X.1252*, each (digital or real-world) entity has a set of characterizing attributes[7]. Each set of attributes whose values allow for *recognizing* an entity in a specific

[3] "property that information is not made available or disclosed to unauthorized individuals, entities, or processes" [Isoi09]. If data about individuals is collected and privacy has to be guaranteed, this requires ensuring confidentiality.

[4] "Any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself" [Syst10].

[5] "[…] the potential impact of a threat and the likelihood of that threat occurring." [Nist06]. We note that to measure risk is often a significant challenge, and different methodologies exist. In practice, risks prevented and risks remaining need to be contrasted with the costs of measures, and a balance should be sought [Nist12].

[6] "[…] an identified natural person or a natural person […] who can be identified, directly or indirectly […]"[Jour16].

[7] „Information bound to an entity that specifies a characteristic of the entity" [Itut10].

context is called identifier[8]. The process of recognizing an entity is called *identification*[9]. Recognizing an entity means *distinguishing* it from others [Itut10, PKLK15]. An entity can have multiple identifiers in the same context as well as different identifiers in different contexts [Itut10]. The entities for which breaches of privacy must be prevented in biomedical research are individuals [PKLK15]. In this context, for a definition of identification, we refer to it as the "process of recognizing an entity by contextual characteristics" [Itut10].

1.1.1 Pseudonymization in Biomedical Research

While anonymous data are not considered personal data in a regulatory sense, pseudonymous data remain personal data [Jour16]. Patient privacy would be best served if all research data were anonymous but there is often the need for follow-up and also the need to combine data from different sources as well as tracing back to patient (e.g. when new cure for a disease is available). This means that the link to a patient's identity may not be destroyed. This leads to an alias or pseudonym[10] which is used to hide a patient's true identity, as long as the data resides in research environments [PKLK15].

Pseudonymity or pseudonymization in biomedical research describes the separation of directly identifying data from genotypic and phenotypic data. Pfitzmann describes pseudonymity as "the use of pseudonyms as identifiers" [PfHa10]. There is a distinction between irreversible and reversible pseudonymity: In this work, we will focus on the latter.

A *pseudonym* is a special type of identifier for which the link to the corresponding entity[11] is kept *confidential*, e.g., by protecting the mapping relation [Itut10]. A newly assigned name of an individual assumed for a specific purpose is a typical example of a pseudonym [Room10]. In biomedicine, alphanumerical pseudonyms are often employed as identifiers

[8] "One or more attributes used to identify an entity within a context" [Itut10]. Identifiers can be *open* or *secret* [Itut10]. The latter means that the binding to the corresponding entity is kept confidential in a specific context. Moreover, the *visibility* of identifiers can differ between contexts. Keeping it *invisible* means that it will not be shown to users [WiJo91].

[9] The process of *recognizing* an entity by a subset of its *characterizing attributes* [Itut10]. The subset is called *identifier* [Itut10] and the characterizing attributes that can cause harm are called *sensitive attributes*. Recognizing an entity also implies *distinguishing* it from others [Itut10].

[10] "An identifier whose binding to an entity is not known or is known to only a limited extent, within the context in which it is used" [Itut10]. We note that this means that a pseudonym is a secret identifier. Further definitions exist, e.g., [Inte00].

[11] „Something that has separate and distinct existence […]" [Itut10].

instead of real names for data protection [Inte00]. This is consistent with Pfitzmann where a pseudonym is “an identifier of a subject other than one of the subject’s real names” [PfHa10].

Pseudonymization adds an additional layer of protection for person-related data. It has been implemented in many projects (e.g. by the UK Biobank [Ukbi07], the Icelandic biobank run by deCode Genetics [HaGS03] and the German National Cohort [00c]) and it has become an important security measure required by laws and regulations. The term “***separation***” plays a central role in various definitions and regulations. The General Data Protection Regulation of the Council of the European Union data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately defines: “data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is **kept separately**”, and in the German Federal Data Protection Act [Fede09]: “characteristics enabling information concerning personal or material circumstances to be attributed to an identified or identifiable individual **shall be stored separately**”. The Italian Personal data protection code sates: “identification data **shall be stored separately** from all other data“ [Repu03]. There are, however, different definitions of pseudonymity and even synonyms for the term itself (including “coding” and “aliasing” [Fede09, Lowr03]). In [Garf15], pseudonymization is defined as “particular type of anonymization that both removes the association with a data subject and adds an association between a particular set of characteristics relating to the data subject and one or more pseudonyms.”

ISO Technical Specification 25237 - “Health informatics - Pseudonymization” also mentions the mechanism of separation: “identifying and payload data shall be separated” [Inte00]. Separation is a popular method in the related literature but there is no description of the data which is to separate. In order to separate data, one needs a minimum of two data sinks. ISO TS 25237 refers to one data sink as “data enabling the attribution […] to an identified or identifiable data subject”. ISO TS 25237 calls the second data sink “other data” or “payload data” [Inte00][LKPK15]. These high-level descriptions make it difficult to distinguish between data or attributes belonging to one or the other sink. ISO 25237 and Pommerening et al. [PDHG14] have addressed but not completely clarified this [PKLK15].

Pseudonymity is of high relevance for biomedical research, and it has been implemented in a series of projects. We will show that current concepts lack a basic requirement analysis and a sound model. We will have a look at the deficits and then present first steps towards such a model [PKLK15].

1.1.2 Disease Networks

The analysis of phenotype-genotype relations improves the understanding of diseases and supports personalized diagnostics and therapeutic efforts in medicine. Detailed medical follow-up data, solid numbers of patients as well as annotated specimens are essential for analyses. To increase the knowledge on diseases, recruitment of patients using pre-defined inclusion and exclusion criteria are crucial as well as networking. In Europe there are a large number of projects funded by European Commission FP7-Health Work Programme [Euro12] dealing with disease networks.

These networks can be categorized into medical research networks like large disease-oriented networks (e.g. for HIV, Diabetes, etc.), centers for common diseases like neurodegenerative or cardiovascular and networks for rare diseases. Most of which are national multi-site networks containing data from thousands of individuals. Their goal is to improve care for patients and improving the understanding of certain diseases. Disease network systems are highly specialized with regards to design and workflow. They focus on specific needs of research and comprise data from different domains (e.g. identification data, clinical data and biospecimen data). Data entry and re-use has to be performed according to the needs of a specific research group. Security and privacy of patient data must be guaranteed as well as laws and regulations for research data must be adhered [Fkoh10].

Several national and international research networks collect identifying data of participants and associated biomedical data. These data are separated into different pools, potentially combined with specifications of valid data flows and further implementation constraints. The German organization “Technology, Methods, and Infrastructure” (TMF) has developed a generic data protection concept [PDHG14] which serves as quasi-standard for German research networks that are affected by these regulations [PKLK15].

1.2 Problem Statement[12]

"Pseudonymization can be seen as a multi-step process: Firstly, data is separated from each other to *prevent re-identification*. Secondly, it is distributed amongst different data stores, which must be *protected from unauthorized access"* [PKLK15]. In the related literature, we discovered the following problems:

Problem 1: No comprehensive analysis of requirements is available for the development of pseudonymized data management systems. Therefore the approach of designing and developing pseudonymization concepts has been of an empirical nature. Measures were introduced which should increase security. That way, "double coding" or "double pseudonymity" has been introduced. The term has been used with different meanings, and it is not completely clear at which process steps and to what entities or data it should be applied. Neither the concept nor its use is sufficiently understood.

Problem 2: Only rudimentary threat and risk analyses have been carried out for previous approaches, covering only single aspects of proposed architectures. There is no guideline for assessing threats and risks in the context of pseudonymized data management in a comprehensive manner.

Problem 3: Neither commonly accepted concepts nor evidence exist to guide the protection of (separate) data repositories. Most solutions distribute data amongst different backend servers, but concrete implementations differ. Some approaches additionally use encryption, others introduce operationally and legally separated storage providers.

Problem 4: The question, which of the measures applied are effective and which properties they have in terms of complexity of implementation and performance impact, has not yet been answered systematically.

[12] Parts of this chapter have been published in [PKLK15]

1.3 Contributions

We consider the contributions of this work as first steps towards a model for pseudonymization approaches with stringent privacy guarantees implemented with comparable and well-defined security techniques. The contributions in detail are as follows:

Requirements on Pseudonymized Data Management

We describe requirements on pseudonymized data management for a common and typical use case: collaborative electronic collection of biomedical research data and further payload data, such as annotations of biosamples. The data themselves may be heterogeneous (structured and unstructured text, images, “omics” data). We will focus on separation and on the management of pseudonyms, and we will restrict the view on data to structured forms as a core element. We derived these requirements from two pseudonymization models ([Inte00, PDHG14] and an overview of standard requirements for GCP-compliant data management in multinational trials [OKCL11]).

Reference Architecture for Pseudonymized Data Management

For the development of our framework, we will use our description of fundamental requirements and then present a high-level architecture of a system that fulfills these requirements. Next, we will provide an overview and a comparison of different technical options for implementing this architecture. Finally, we present a comprehensive implementation framework for pseudonymized data management and show its feasibility by describing how a national [BGLK12] and an international [KLKB12] research network have been implemented. We consider the methods presented here technical blueprints, which can be used to implement different pseudonymization schemes.

Risk and Threat Analysis for the Reference Architecture

For our reference architecture, we performed a thorough risk and threat analysis according to the STRIDE/LINDDUN methodology [DWSP11, MHLO14]. We were herby able to systematically model threats and assess associated risks. Furthermore, we identified countermeasures[13] to mitigate the risks. Regarding the related work on pseudonymization concepts and solutions, we are the first to present a sound and comprehensive threat and risk analysis for our solution.

[13] "Actions, devices, procedures, or techniques that meet or oppose (i.e., counters) a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken" [Syst10].

2 Requirements on Pseudonymized Data Management[14]

From a functional perspective the basic idea of biomedical research data collection is to provide a system for capturing and managing research data. Data needs to be managed for physical entities, like patients and biosamples, and also for events like encounters. Encounters are typically managed as electronic case report forms. Slightly simplified, such a system only needs to manage one logical type of entity: documents describing instances of a specific physical entity or event. Additionally, the relationships between the entities must be managed, typically in a tree-like structure. Different physical entities are represented by different types of logical documents. Often the entity real-world “subject” (person, biosample) is modelled explicitly because it represents the root element of a document structure. The document associated with an instance of this entity contains a subject’s master data. The basic idea of any pseudonymization concept aims at distributing documents across databases depending on their type (i.e. the entity or type of entity described by them). In order to gain solid numbers for statistical analyses, studies often have to be set up in a multicentric manner. Depending on the nature of a study, solid numbers in terms of patients are only achievable by cooperation and networking. Medical research tends to be increasingly organized as research associations, mostly with a highly disease specific focus [PDHG14]. Pommerening names in [PDHG14] the following scenarios for data collection in medical research databases:

- Establishment of new diagnostic and therapeutic options
- Standardization and optimization of treatment
- Forming new hypotheses as a basis for controlled clinical trials
- Analyses of genetic causes of diseases
- Identification of precautionary measures
- Exploration of psychosocial consequences of diseases
- Recruitment of eligible patients for studies or trials

[14] Parts of this chapter were published in [LKPK15] and [Fkoh10].

2.1 Methods

In order to get a conceptual basis, we have started our work with an analysis of two comprehensive concepts that are most cited in the relevant literature in the context of this work ([Inte00, PDHG14]). While ISO/TS 25237 [Inte00] is an international standard by definition, [PDHG14] is being considered a guideline which integrates a set of quasi-standard requirements and best practices in Germany. From these two sources, we compiled a set of fundamental requirements for pseudonymized data management. Functional requirements were taken from related work on electronic data capturing systems and complemented with results from our own analyses[15]. Ohmann et al. presented in [OKCL11] a list of requirements, that were identified in several national and international standards and publications in order to define standard requirements for GCP-compliant data management in multinational clinical trials. As a basis for this list, the following publications are important: ISO 27001 Information Security Management Specification [Iso16], EU Directive for the implementation of GCP 2001/ 20/EC [Euro01], EU Directive 95/46/EC [COEC12], EU Directive 2005/28/EC [Euro05], Computerized Systems EudraLex - Volume 4 - Annex 11 [Euro08a], EMEA Reflection paper on expectations for electronic source documents used in clinical trials [Gcpi07], ECRIN deliverable D10 - GCP-compliant data management in multinational trials [Euro08b], Good practice for computerized systems in regulated GXP environments PIC/S Inspectors Guide [Phar07], Good Clinical Data Management Practice - Version 4 - of the Society for Clinical Data Management [Soci05], Data and Information Management Systems Project (DIMS) System Standards of UKCRC/NIHR (UK) [Is09], IT-Grundschutz - Methodology of the Bundesamt für Sicherheit in der Informationstechnik (BSI) [Bund08], FDA - Guidance for Industry - Computerized Systems Used in Clinical Trials [Fda00] and 21 CFR Part 11 [FlKe02][16]. Although Ohmann et al. were focused on a slightly different use case, the identified requirements apply in our case as well. From Ohmann's collection of requirements, we extracted those which we considered relevant in our context.

[15] Parts of this section were published in [LKPK15]

[16] The list comprises a few other national publications and project results which we have not listed here.

2.2 Results

In this section, we will describe a set of requirements that (1) are specified in pseudonymization concepts [Inte00, PDHG14] and thus define our implementation of pseudonymity, and, (2) result from the influence of data separation on the system in terms of support for our requirements. The selected requirements are broad enough, to result in a generic solution for different types of data. To describe the central aspect of “separation”, we have started by focusing on the data layer. Typically, however, information systems are described by further layers, comprising an application and a presentation layer [ACKM04], which support (and to some degree model) real-world processes [DaRK98]. We substructured the requirements following [OKCL11] into “data entry”, “data management“, “physical security”, “logical security”, “business continuity”, software development”, “database management”, “export” and “reporting” requirements. In [LKPK15], we identified the following roles in our system:

- ***Physician***: Responsible for examination and treatment of a patient as well as data entry.
- ***Nurse***: Member of the physician’s team, responsible for data entry.
- ***Researcher***: Performs research using provided data or biomaterial. Responsible for feedback to physician about results of analyses on data or biomaterial.
- ***Lab personnel***: Member of a researcher’s team. Responsible for feedback to physician about results of analyses on data or biomaterial.
- ***Data manager***: Responsible for access rights assignment, quality assurance and quality control of patient data entries and query management.
- ***Study nurse***: Responsible for quality assurance and quality control of patient data entry and query management.
- ***Server administrator***: Responsible for administration of servers.
- ***Application administrator***: Responsible for administration of clinical data management application.
- ***Software developer***: Responsible for development and maintenance of clinical data management application.
- ***Data protection officer***: Responsible for control of security infrastructure, user trainings and user information.

2.2.1 Data Entry Requirements

The following requirements apply to the roles physician, study nurse and lab personnel, as described in [LKPK15]. They represent the users of the system that perform remote data entry.

Table 1. Data entry requirements

Req.ID	Requirement	Description
DER-1	Data collection	The system shall support collection of research data and metadata about associated entities in electronic forms or documents (eCRFs). [ASWS08, Fkoh10, JAHC09, MOFH12, OKCL11].
DER-2	Data structuring	The system shall maintain links between electronic forms or documents (eCRFs) [ASWS08, Fkoh10, JAHC09, MOFH12, OKCL11].
DER-3	Follow-up	The system shall provide follow-up data entry. In scenarios with follow-up data collection additional information about a patient or proband needs to be entered [ASWS08, Inte00, PDHG14].
DER-4	Integrated view	The system shall support integrated views of different forms or documents between which an association exists [BBHP15, KLKB12, MOFH12].
DER-5	Data quality and validation checks	The system should support checks for validation of eCRF data entry [ASWS08, OKCL11].
DER-6	User friendliness of eCRFs	eCRFs shall follow consistent design principles and shall have simple and clear instructions [ASWS08, OKCL11].
DER-7	Standard web browser client	The system shall not need any additional client software installation apart from a standard web browser [Fkoh10].

2.2.2 Data Management Requirements

The following requirements apply to the roles data manager, physician, study nurse and lab personnel. These roles perform management of single entries of a user‘s site as well as the management of all entries by, e.g., a data manager.

Table 2. Data management requirements

Req.ID	**Requirement**	**Description**
DMR-1	Management of missing eCRFs	The System shall report missing data [OKCL11].

DMR-2	Design of eCRFs	eCRF design process shall be documented and reviewed [OKCL11].
DMR-3	Cross-disciplinary team	The clinical data management application and eCRF design and development shall be performed inter-disciplinary [OKCL11].
DMR-4	Standardized questionnaires/ instruments	Standardized/validated questionnaires and scales shall be used, if possible [OKCL11].
DMR-5	Integrity and completeness	The system shall provide means to maintain validation of data completeness and integrity. [BBHP15, Fkoh10, OKCL11].
DMR-6	Query creation and tracking	The system shall support the creation and tracking of queries. Queries shall be supported for patients and for biospecimen [ASWS08, OKCL11].
DMR-7	Locking of data	The system shall support the locking of single and multiple data entries [PDHG14].
DMR-8	Deletion of data	The system shall support the deletion of single data entries [PDHG14].
DMR-9	Distributed CRUD[17]	The system shall implement distributed create-, read-, update- and delete operations on top of a set of distributed databases [Inte00, PDHG14].
DMR-10	Re-identification[18]	The system shall support the de-pseudonymization of data in order to re-identify single data subjects [ASWS08, Fkoh10].
DMR-11	Usability	The system should adhere to well-established usability guidelines [Beva97].
DMR-12	Patients attending multiple sites	The system shall support the detection and reconciliation of patients that attended multiple sites [Fkoh10].
DMR-13	Feedback of researchers	The system shall provide researcher means to enter their results on data- or biomaterial analyses into the system [Fkoh10].

[17] The two concepts [Inte00] and [PDHG14] define pseudonymization of a dataset as a separation into subsets containing different types of data. The records within these subsets are stored in different locations and they are interlinked with identifiers. As suggested in [LKPK15] and [PKLK15] Data collection and management can be modeled as a set of *CRUD* operations on documents: (1) *Create*: creates a new document, (2) *Read*: provides a view of the data contained in one document or a list of other documents related to one document. (3) *Update*: provides a view of the data contained in a document while allowing updating its content. (4) *Delete*: deletes a document. The above aspects are covered by the requirement *DMR-9*. As a result of *DMR-9*, operations on documents must be performed across different data pools.

[18] *DMR-10* plays an important role in re-contacting a patient or proband based on the informed consent. To re-contact a patient is often a mandatory process for research networks. The network needs a patient's consent in order to do this (*DMR-15*). Reasons for re-contacting are a reminder for a follow-up visit, the feedback of findings or the recruitment for other projects. Re-contacting is performed by the personal physician or staff [PDHG14].

DMR-14	Adherence to patient consent	All operations performed on the data must adhere to the patient informed consent [Inte00, PDHG14].
DMR-15	Re-contacting a patient	The system shall provide means to re-contact a patient in case of relevant events. The system shall also remind the persons responsible for data entry, when a patient needs to be re-contacted (e.g. reminder for yearly follow-up) [PDHG14].

2.2.3 Physical Security Requirements

The following requirements are directed towards peripheral hardware devices and locations where the devices are operated. These requirements describe physical protection.

Table 3. Physical security requirements

Req.ID	Requirement	Description
PSR-1	Physical access restrictions to client and server hardware	Physical servers location must be in a dedicated and locked room with unescorted access only for specified persons [OKCL11].
PSR-2	Secured power supply	The server’s power supply should guarded by a UPS [OKCL11].
PSR-3	Encryption of non-physically secure data	Storage of patient data should only be allowed on protected servers. Patient data stored on mobile devices must be encrypted [OKCL11].
PSR-4	Server failure - response	Alerts on server failure should be sent automatically to relevant personnel [OKCL11].
PSR-5	Controlled environment	Servers should be located in a temperature controlled environment [OKCL11].
PSR-6	Server room/building linked to response centers	The server room and the building should be linked to a central response center and have an alarm for emergencies [OKCL11].
PSR-7	Hazard control - fire alarms	The server room should have permanently monitored heat and smoke alarms [OKCL11].
PSR-8	Hazard control - fire response	The server room should have automatic fire response [OKCL11].

PSR-9	Physical separation and spatial distribution of data[19]	The system shall support the hosting of data on different backends and different physical machines with different host names [ASWS08, PDHG14].
PSR-10	Separation of powers and duties	Servers hosting the different backend systems must be spatially and organizationally separated. This includes as well different rooms and staffing for the separated backend servers [PDHG14].

2.2.4 Logical Security Requirements

The following requirements comprise logical protection mechanisms for data and transmission of such as well as best practices regarding staffing and standard procedures.

Table 4. Logical security requirements (I)

Req.ID	Requirement	Description
LSR-1	Security management system	"Regular reviews of IT security systems, practices and documentation, […], should occur as part of an ongoing Security Management System" [OKCL11].
LSR-2	Commitment to data protection	A data protection officer shall keep watch over relevant security policies and trainings [OKCL11, PDHG14].
LSR-3	External firewalls	The system should be secured by external firewalls [ASWS08, OKCL11, PDHG14].
LSR-4	Encrypted transmission	All data transmitted over the internet must be encrypted with state of the art encryption technology, e.g., TLS with server certificates, SHA-256 encryption for files [ASWS08, OKCL11, PDHG14].
LSR-5	Server admin role	A servers admin role should be in place with appropriate password protection and only limited to certain individuals [OKCL11].
LSR-6	Admin password management	The administrator password should comply to password policies and have an off-site emergency backup [OKCL11].
LSR-7	Server maintenance	"Necessary patches and updates should be identified and applied in a timely but safe manner to: the operating system, anti-malware systems, backup systems and major apps (e.g. Clinical DBMSs, Web servers, Remote Access systems, etc.)" [OKCL11].
LSR-8	Commitment to information security	The unit and/or parent organization should be committed to information security and advocate for according policies, trainings and dedicated roles [OKCL11, PDHG14]
LSR-9	Internal firewalls	The system should be protected by internal firewalls [OKCL11].

[19] The hosting of different backends on different physical machines with different host names is summarized in *PSR-9* "physical separation and spatial distribution of data".

LSR-10	Security testing	Frequent security testing should be conducted and documented [OKCL11].
LSR-11	Intrusion detection with traffic monitoring	Traffic should be monitored and suspicious behavior and/or anomalies identified and investigated [OKCL11].
LSR-12	Logical access procedures	Standard operating procedures with policies for access control should be in place [OKCL11].
LSR-13	Access control management	Access should be differentiated and managed using role-based access control mechanisms [ASWS08, OKCL11, PDHG14].
LSR-14	Granularity of access	Access control granularity should comply with the Need-to-Know Principle [Fkoh10, OKCL11, PDHG14].

Table 5. Logical security requirements (II)

Req.ID	Requirement	Description
LSR-15	Password management	Password management policies should be in place for all users [OKCL11].
LSR-16	Desktop lockout	Desktop logins should perform an automatic logout and/or specified shut down period [ASWS08, OKCL11].
LSR-17	Review of data and system access rights	“Access rights […] should be regularly reviewed, changes to access requested and actioned according to defined procedures, by designated individuals, with records kept of all rights, when granted, why and by whom.” [OKCL11].
LSR-18	Data security	All authorized personnel involved will keep data secure and confidential at all times [OKCL11, PDHG14].
LSR-19	System security	Data shall only be accessible to authorized individuals [OKCL11].
LSR-20	Client-side re-combination[20]	Systems need to be designed in a way that the reconstruction of the logical global dataset can *only be performed at the client-side* to reduce the number of attack vectors [PDHG14].
LSR-21	Confidentiality of internal identifiers	Clients must not be able to learn the pseudonymous identifiers used in the distributed databases. Pseudonymized datasets may only be joined by authorized users [PDHG14].

[20] Pommerening et al. [PDHG14] added the requirements *LSR-20* and *LSR-21* on application-layer. *LSR-20* describes the client-side re-combination of the logical global dataset to reduce the number of attack vectors.

LSR-22	Two-tier pseudonymization[21]	The system shall provide support for two-tier pseudonymization, implemented with an additional mapping service [Inte00, PDHG14].
LSR-23	Restriction of data access	The system shall support a site-based view where a physician and staff have access restricted to data of their site. The system shall support researchers having access restricted to data concerning biospecimen. Furthermore, the system shall support data managers having access to patients medical data without having access to identification data [OKCL11, PDHG14].
LSR-24	Logging procedures	Every request to a system must be logged in a separate file [ASWS08, OKCL11, PDHG14].

2.2.5 Business Continuity Requirements

In case the system should be unreachable to its users, a Business Continuity plan should assure established procedures for restoring the system to its last working point and comprise offline procedures to maintain business continuity.

Table 6. Business continuity requirements

Req.ID	Requirement	Description
BCR-1	Business continuity Plan	A plan for Business Continuity should exist in case of emergency [OKCL11].
BCR-2	Backup policies	Backup policies and procedures for restoring and testing must exist [OKCL11].
BCR-3	Backup frequency	Backups must be made at a specified frequency that ensures minimal loss in case of emergency [ASWS08, OKCL11].
BCR-4	Backup storage	Backups should be stored fire proof [ASWS08, OKCL11].
BCR-5	Off-site archiving	Backups should be also stored off-site [OKCL11].
BCR-6	Back up - environment	“The server / DBA environment (groups, log-ins, jobs etc.) should be captured and restorable” [OKCL11].
BCR-7	Maintainability	The system should allow for centralized installation and maintenance [Fkoh10, MOFH12].
BCR-8	Virtualization	The system should be hosted on virtual servers [LKPK15].

[21] Our aim is to provide a generic solution and both pseudonymization concepts require two-tier pseudonymity (*LSR-22*) when some types of data are involved (e.g. links to biospecimen). Therefore, a generic system needs to implement *two-tier pseudonymization,* which must be realized with an additional mapping service.

2.2.6 Software Development Requirements

The software development requirements are not in the scope of this work. We therefore picked only the most important in our context. For a comprehensive list of software development requirements, we refer to Ohmann et al. [OKCL11].

Table 7. Software development requirements

Req.ID	Requirement	Description
SDR-1	Documentation of in-house software	Software documentation should adhere to state-of-the-art guidelines and best practices [OKCL11].
SDR-2	Open-source software	The system shall be developed using open-source software components to avoid license costs and software sharing as well as restrictions on publication [Fkoh10].

2.2.7 Database Management System Requirements

The following requirements apply to database management systems and represent only the most important ones in our scenario. For a comprehensive list of database management system requirements, we refer to Ohmann et al. [OKCL11].

Table 8. Database management system requirements

Req.ID	Requirement	Description
DBR-1	Development and production Instances	The system offers a development and production environment [ASWS08, OKCL11].
DBR-2	Audit trail	All transactions to the database are recorded in an audit trail [ASWS08, OKCL11].

2.2.8 Export and Reporting Requirements

Export and reporting are common functionalities in the context of our solution. There is a huge demand for exports and reports on a regular basis for quality assurance and project management purposes. For a comprehensive list of export and reporting requirements, we refer to Ohmann et al. [OKCL11].

Table 9. Export and reporting requirements

Req.ID	Requirement	Description
DBR-1	Report access control	Access control mechanism should guard access to reports [OKCL11].
DBR-2	Single subject data export	The system should support “to examine and export a full record of a single subject’s data (excluding personal identifying data)” [OKCL11].
DBR-3	Standard reports	Standard reports should be available to authorized personnel [ASWS08, OKCL11].
DBR-4	Data export procedures	Standard operating procedures as well as dedicated policies for exporting data should exist [OKCL11].
DBR-5	Purpose recorded	All data transfers and purpose of transfer should be documented and recorded [OKCL11].
DBR-6	Assuring security	Data Use Agreements must be in place that cover aspects of maintaining protection of transferred data [OKCL11][LKPK15].

2.3 Discussion

2.3.1 Principal Results

As users interact with the system on presentation layer, these requirements also relate to our functional requirements. Re-identification (*DMR-10*) is a core element related to (reversible) pseudonymity. On the real-world level, re-identification means revealing the hidden identity of a subject. For re-identification *the separation* between identifying data and payload data needs to be reversed (at least on the presentation layer). There are several activities of data management which require re-identification. Also, in the context of follow-up data collection re-identification must be performed in order to retrieve the corresponding dataset. The requirement *DMR-10* is implied by functional requirement *DER-4* (Integrated view).

The system should *adhere to well-established usability guidelines (DMR-11).* While often technical details of solutions are not described in publications, there are reports on systems in which the linkage of different data subsets must be performed manually, i.e., by copying and pasting an identifier displayed by the interface of one application into the interface of another application [LaBÜ15]. This process is time-consuming and error-prone [GoNT08]. Furthermore, implementing a consistent user interface for several distributed systems while ensuring a continuous workflow means that there should be no need for users to separately authenticate on the multiple systems involved.

Data separation will ultimately lead to more complex system architectures which in turn may be more difficult to maintain. We have therefore added *(BCR-7) "*maintainability". Supporting a large set of users that are distributed geographically requires methods for *centralized installation and maintenance*. Integration into enterprise security architectures of clinical or research facilities will bring up challenges like institutional firewalls or software installation policies. Furthermore, as pseudonymization architectures consist of several servers where the sub-components are deployed, there is also a strong emphasis on maintainability.

As basic requirements we identified the support of longitudinal studies (follow-up, *DER-3*), the identification of patients moving between study centers (*DMR-12*), the patient information in case of relevant events (*DMR-10*). Furthermore, the regional and national collection, storage, management of identification data, clinical data, specimen- and image associated data. Web-based structured forms are needed for data entry (*DER-1*). Any installation of additional client software should be avoided to assure platform independence, little maintenance effort and broad availability (*DER-9*). The integration in enterprise security architectures has constraints like company firewalls, software installation policies. Access rights are based on informed consent (IC), institutional review boards (IRB) and country-specific laws and regulations. The specific research workflow from patient admission to release must be supported and system must reflect the permissions given by a patient through informed consent. In case of a revocation of consent by the patient, there has to be a way of data-deletion or anonymization (*DMR-8*). Patients always have right to withdraw their consent. This includes that a research alliance may lose the right to store, process or use the data provided by the patient. It depends on the patient's consent what exactly needs to happen in case of consent revocation. The data may be locked, anonymized or deleted (*DMR-7, DMR-8*) [PDHG14]. The system has to keep protocol of all data accesses and modification operations to assure confidentiality and integrity (*DBR-2*).

As required by *PSR-9* and *PSR-10* a pseudonymization process for data and specimen must be in place. In addition, data must be separated spatial and organizational. The re-labeling of specimen-tubes in the collection phase should be avoided, because it is likely to cause mistakes. Furthermore the security of patient data is related to consent and IP aspects. A role based access control (RBAC) is needed to support roles like physician, researcher and lab personnel with different rights and permissions (*LSR-23*). Site-based views should ensure that only the personal physician and authorized team-members can access identification data of patients (e.g. names, addresses) (*LSR-23*). Reliability, robustness, performance and user-friendliness are important features that are obligatory (*DER-8, BCR-7*). It is important to make the software available to other workgroups or networks therefore it should be scalable, extensible and a generic, re-usable solution without license costs (*SDR-2*).

According to [PDHG14] the handling of sensitive medical data in a research network requires a carefully designed role-based access control mechanism. It needs to be structured along the "nee-to-know" principle. The basis for access control is formalized in the study protocol of a research alliance. Access rights are assigned to roles, and roles in turn are assigned to user accounts. The person that assigns these access rights to roles and the latter to a user's account is the data manager (*LSR-13, LSR-14*).

The collection of data is an error prone process. Data are often incomplete, have typos, were collected for another purpose or have deviations in coding. Quality assurance processes must be in place that data are prepared in a way that they are consistent with the required format and specification (*DMR-5, DER-5*). Data can be sanitized at data entry, e.g., they may be checked for completeness and plausibility depending on the EDC system used. In case of doubt or inconsistencies the data source is requested for a revision of the entry (source data verification) [PDHG14].

Reporting of data is an important functionality for statistical analyses. The data manager can generate simple reports for quality assurance, benchmarking and formulation of hypotheses (*ER-3*). These reports need to conform with data minimization and should be anonymous [PDHG14].

The sharing of data with researcher represents the final step of a research network's purpose. Data may be shared with other research projects or be reconciled with other cohorts or data pools. The sharing of data is also subject to a patient's informed consent and needs as well to be conforming data minimization. Data are shared in form of an exported dataset according to the researcher's need (*ER-4*) [PDHG14].

The risk or unauthorized re-identification must be minimized. Therefore, the principle of separation of powers and duties is in place (*PSR-9*). This includes the physical separation of data as well (*PSR-10*). It results in a combination of technical and organizational protection measures. The technical measures, e.g., access control, encryption and logging procedures (*LSR-24*) are supplemented by additional organizational measures, e.g., standard operating procedures, clearly defined responsibilities and four-eyes-principle. Data transmission shall always be encrypted with state of the art encryption (*LSR-4*) [PDHG14].

2.3.2 Comparison with Prior Work

In [PKLK15], [Fkoh10] and [LKPK15] we systematically examined the requirements on pseudonymized data management, which represent the cornerstone of this work. Bialke et al. present in [BBHP15] a short overview of functional and non-functional requirements of their system. The requirements presented by Angelow et al. in [ASWS08] address central security issues. They were set up for a central biosample and data management solution. However, it is only a short overview over important security requirements, lacking central considerations regarding the characteristics of pseudonymization. The most elaborate approach is the work by Ohmann et al. [OKCL11], although it has a different focus. Since our system does not need to be GCP-compliant, many of the formulated requirements do not apply in our case. It still represents a comprehensive overview for the intended use case.

2.3.3 Limitations

We left out requirements that are specific to clinical trial management systems, because our system does not require to be GCP compliant. Furthermore, we did not perform a prioritization of requirements. Our selection of requirements represents the very essence of our system's functionalities and workflows. Therefore all are requirements mandatory and as a result, they need to be implemented.

There are many articles that focus on application-level aspects of pseudonymization and do not describe technical details about the information systems that manage these data and implement the described processes [ClMM03, IvGr96, MaSc12, MoCM03, NoLL07, WyMi03]. Some articles on pseudonymized data management focus on different real-world applications than this thesis. This results in other functional requirements. An important group consists of approaches in which re-identification is only supported as an exceptional procedure [HeKN11, Iaco07, KSMM05]. Articles on pseudonymization in which patients control access to their data (e.g. via smart cards [NeHe11, RiGN08]) were not examined here.

2.3.4 Conclusions

We have presented the first comprehensive overview of requirements for medical research networks. This should serves as a basis for the design of a reference architecture and ultimately for the development of a generic solution for pseudonymized data management. Both will be presented in the next chapter.

3 A Reference Architecture for Pseudonymized Data Management[22]

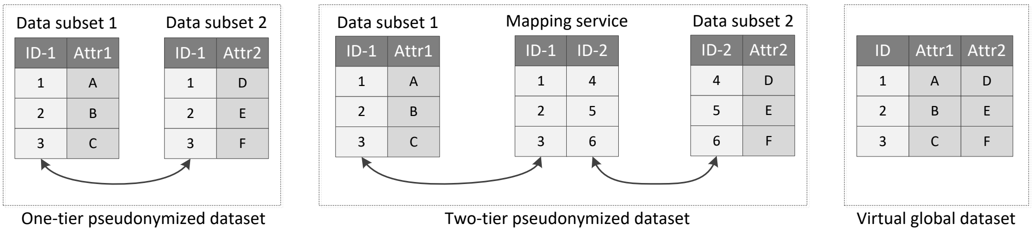
Pseudonymization is most important during data collection and management phase as it allows mitigating identifiability threats while enabling the management of individual-level microdata and supporting all required operations. Furthermore, pseudonymization techniques support longitudinal studies, which require the ability to re-identify a subject during a follow-up visit. Additionally research systems require the ability to re-associate the collected medical data to identifying data, e.g., in case of adverse events or new therapeutic options, which is also possible using pseudonymization techniques. When releasing data, pseudonymization can be used to separate nominal and demographic data from research data to derive a de-identified dataset. As pseudonymity is a reversible process it is of relevance when re-identification is needed for follow-up or when a disease is detected which may result in a direct threat to a patient or his/her environment. In addition, if biosamples must be returned, the use of (temporary) pseudonyms is essential for their management.

In this section we describe a generic solution for implementing pseudonymized data management for a common and typical use case: collaborative electronic collection of biomedical research data and further payload data, such as metadata about associated entities (e.g. biospecimen). We will include pseudonymous identifiers for associated biospecimen. We will not go into depth concerning entities managed in biobanks, but present a generic solution that is based upon requirements that allow for managing a broad spectrum of data. Furthermore, we will not address data resulting from imaging or genetic analyses. We will focus on structured forms as a core element, on separation, and on the management of pseudonyms. This also means that we will restrict the view on data maintaining links to structured forms as a core element.

We will describe a generic solution which offers a seamless integration and transparent access of multiple distributed data stores into a holistic information system with

[22] Parts of this chapter were published in [LKPK15] and [Fkoh10]. The concerned sections are marked accordingly.

transparent user access. It was designed for an integrated management of pseudonymized data. The solution has successfully been used to implement international and national research networks and can serve as a blueprint for the development of future systems.



ID-1	Attr1
1	A
2	B
3	C

ID-1	Attr2
1	D
2	E
3	F

ID-1	Attr1
1	A
2	B
3	C

ID-1	ID-2
1	4
2	5
3	6

ID-2	Attr2
4	D
5	E
6	F

ID	Attr1	Attr2
1	A	D
2	B	E
3	C	F

Figure 2. Data-layer separation: examples for pseudonymized datasets[23]

As separation of data is a core characteristic of pseudonymization, we illustrate it by Figure 2, showing two different options. Here, *Attr1* can be considered "identifying", whereas *Attr2* are "payload". Option 1, which we call "one-tier pseudonymized" has been used in trials for decades, the "two-tier pseudonymized" approach is more recent and it is recommended by ISO 25237 [Inte00] and by Pommerening et al. [PDHG14]. According to the terminology of ISO 25237, attributes *A*, *B*, *C* are considered as "identifying" whereas *D*, *E*, *F* are considered as "payload". Two-tier pseudonymity describes a dataset that is interlinked with identifiers. In two-tier pseudonymized architectures, each data sink has a separate *namespace* for identifiers. Dedicated identifiers of other namespaces are mapped via a mapping table or mapping service.

We will refer to this (simplified) data-centric view of Figure 2 throughout this chapter. As noted above, we will focus on separation without further address the specifications of addressing definitions of terms like "identifying", "quasi-identifying" or "indirectly identifying"[24] data and "payload" [Inte00], but recommend clarification by further work. Pommerening et al [PDHG14] have introduced additional types of data: 1) identifying data, 2) medical- or clinical phenotype data, 3) data associated with the management of biospecimens, and 4) data resulting from the analysis of biospecimens.

[23] Cf. [LKPK15]

[24] Data that can identify a single person only when used together with other indirectly identifying data [Inte00].

3.1 Methods

Based on the identified requirements on pseudonymized data management (cf. chapter 2), we wanted to describe a solution for pseudonymity architectures. Therefore, the next step was to design a system architecture fulfilling the requirements identified. We examined several architectural designs and created an overview of technical options for implementation. With this we were able to implement a generic solution. Its feasibility has been demonstrated in research networks of which we will shortly describe ([BGLK12, KLKB12]). As regulations are not precise enough to directly provide an implementation guideline, we based our work on the pseudonymization concepts [Inte00, PDHG14]. Both do not sufficiently specify attribute types in the data pools, and both contain only rudimentary process descriptions, so we had to add specifications based on user feedback[25].

[25] Parts of this section were published in [LKPK15]

3.2 Results

3.2.1 Architectural Options

In [LKPK15], we have analyzed techniques for integrating distributed databases into one interface. In Figure 3 this design space is illustrated. In the following, we will go into detail regarding technologies to build modern web applications.

Loose coupling: Describes presentation-layer integration performed by a client. Here, a user has to access different interfaces of separate systems in a sequential order. These interface can be displayed next to each other or be embedded (potentially combined with methods for context management such as HL7 CCOW [Hl00]). Loose coupling only supports a very limited handover of data between single interfaces (*interface-to-interface communication*). CRUD-Operation (create, read, update, delete) on entities need to be manual operations, performed on multiple interfaces of the systems over which the data of an entity is spread. Identifiers must also be copied manually between the systems, to maintain consistency. This workflow is highly error-prone and inefficient and may cause massive data quality issues. A user's acceptance of such a system may also be very low due to the different systems involved potentially have different workflow and interaction design (look-and-feel) [LKPK15]. The majority of articles we found in the literature regarding pseudonymized data management are based on the principle of loose coupling [EWAG07, FaPo05] and [DDGH10].

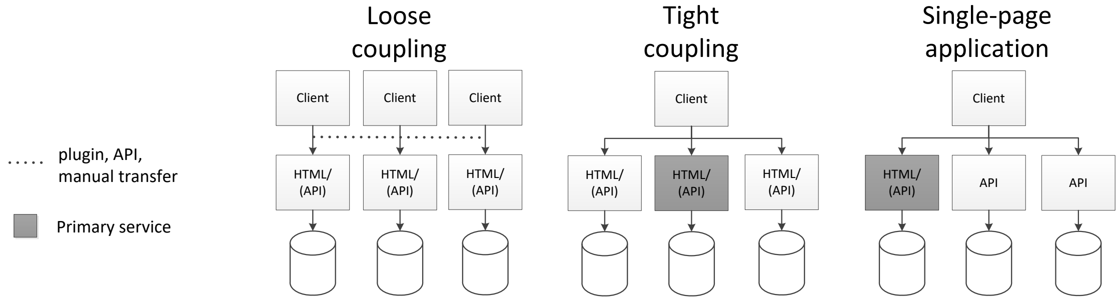


Figure 3. Design space for distributed data management[26]

Tight coupling: Describes an integrated access to several data pools (shown in Figure 3). Here, the *primary service* is a special endpoint that provides the main application and

[26] Cf. [LKPK15]

integrates user interfaces of other endpoints on presentation layer. Each endpoint can also provide a programming interface to access its data. Compared to loose coupling, the *primary service* needs to manage data from different data pools and also different domains. This results in a much more complex business logic for the *primary service* because of the orchestration of access and interaction between the separated endpoints [LKPK15].

***Single-page application*:** Describes a design with a single graphical interface provided by one endpoint. This endpoint is able to access data of all other endpoints via dedicated interfaces (cf. Figure 3). The data pools need only provide very light weight data access interfaces (i.e. *Web Application Programming Interface*) which results in a small business logic to implement. Suitable candidates would be *Representational State Transfer* (REST) interfaces or *Web Services* (WS). For data exchange in between these interfaces *JavaScript Object Notation* (JSON) or *Extensible Markup Language* (XML) could be used. For implementation a client-side *Model-View-Whatever* (MVW) framework (e.g. *AngularJS* [00d] or *Backbone.js* [00e]) would be possible [LKPK15].

3.2.1.1 High-level Architecture

Based on our requirements analysis and the architectural options outlined in [LKPK15] and [PKLK15], we designed a high-level system architecture. We wanted reduction of installation efforts and ensuring compatibility with other systems. Therefore, we implemented a *web-based system* that can be accessed with a broad spectrum of browsers by adhering to web standards. Hereby, the distribution of updates to our software was simplified (*BCR-7*)[27].

Regarding the architecture, a loosely coupled system was not an option due to the issues mentioned above. The problem with Single-Page Application (SPA) is that the frameworks and technologies are not supported by legacy web browsers. Therefore, we made a *tightly coupled design* that fulfills the requirement of seamless integration (*DMR-10*) as well as good usability (*DMR-11*).

The requirement (*LSR-20*) describes the reconstruction of the dataset at client side. It is fulfilled by the usage of *client-side mashup-techniques*. Here, data from different origins is displayed integrated within a user's local browser. The support of multi-tier pseudonymity

[27] Please cf. chapter 2 for referenced requirements.

(*LSR-22*), was realized by a *mapping service*. It maps pseudonymous identifiers of the involved systems to each other. The *client-side mashup* makes sure that distributed data is only combined at the client side. Furthermore, data is exclusively delivered to the clients. There is no data exchanged in between the data pools. Clients should also not be aware of pseudonymous identifiers used (*LSR-21*). This is realized in the data pools before data exchange by *substitution of identifiers with temporary identifiers.* Ensuring consistency and the support of common database operations (*DMR-9*), is realized by the management of data in several distributed *relational database management systems* (RDBMSs).

The above design decisions have massive implications to a system. For having a fluent workflow, *Single-Sign-On (SSO)* must be in place (*DMR-11*). Furthermore, the *Same-Origin-Policy (SOP)* of common web browsers complicates client-side mashups by restricting the retrieval of data from multiple origins. The SOP states that “only the site that stores information in the browser may later read or modify that information”[JaBo06]. Cross-domain communication is hereby prohibited. However, cross-domain communication is required to recombine a dataset from different data pools, stored on different physical servers (PSR-9). In addition, mapping services must be integrated. As described earlier, pseudonymous identifiers are substituted with temporary identifiers before data leaves a data pool. To enable the reconstruction of the separated data the temporary identifiers must be synchronized between the different data pools. Therefore, a *secure server-to-server communication channel* is needed that cannot be accessed by a client [LKPK15].

3.2.1.2 Implementation Options

To implement the described architecture, different methods can be utilized. We will compare the different possibilities for the realization of the crucial elements of the system which are:

- Client-side web mashup
- Single-sign-on mechanism
- Secure server-to-server communication channel

3.2.1.2.1. **Web Mashups**[28]

[JaWa07] defines a mashup as "a website […] that seamlessly combines content from more than one source into an integrated experience". In this context, the challenge is the implementation of a mashup from data that is stored in different physical locations. The data need to be accessed via multiple interfaces provided by endpoints different domains. This in turn raises problems with the *Same-Origin Policy* (SOP). As described earlier, this security mechanism prohibits cross-domain communication. The idea was the protection of a user's privacy by prevention of user behavioral tracking. The corruption of a user's actions is also prevented by SOP and in addition websites cannot perform transaction on the user's behalf [JaWa07]. Therefore, the SOP allows scripts to only modify the contents of a website of the same origin (i.e. from the same domain). The article by De Ryck et al. shows a variety of state-of-the-art mashup techniques [RDDP12]. Most used components in today's mashups are *HTML Frames*, *PostMessage*, *XMLHttpRequest* (XHR) and *JSON with Padding* (JSONP) [RDDP12]. However, the SOP can be circumvented by only a few of them. In the following we will explain the techniques in detail.

HTML Frames**:** An HTML-frameset subsumes a group of HTML-frames. The frame contents are dynamically loaded. Each frame and its contents are independent form the other frames in a frameset. With the introduction of IFrames (inline frames) in HTML 4.0. it was possible to embed HTML-frames in the body of other HTML-documents. Iframes and Framesets can be employed to display information from different origins. However, they do not support any interaction in between [LKPK15].

PostMessage**:** HTML postMessage allows for cross-domain communication. It enables scripts to send data to HTML Frames or windows of arbitrary origin. This technique is available with the introduction of HTML 5. The idea is that the recipient of a message verifies that the message origins from a valid/authorized sender [SoSh13]. Legacy browsers do not support this technique [LKPK15].

XMLHttpRequest (XHR)**:** Most web browsers support the XHR API. It allows to send HTTP requests to a server. The server answers in return with XML-, TEXT/HTML- or JSON-formatted data. XHR API can be implemented web scripting languages (e.g. JavaScript). Cross-domain communication can be supported by XHR if a client supports Cross-Origin

[28] Cf. "Web mashups" in [LKPK15]

Resource Sharing (CORS). Since CORS is a W3C recommendation from early 2014, it is not supported by legacy browsers [LKPK15].

JSON with Padding (JSONP)**:** JSONP is a mechanism which uses the src-attribute of an HTML script tag, to which the SOP does not apply. The endpoint from which data should be loaded is defined in the src-attribute of the script tag. The endpoint embeds the requested data into a callback function of a local JavaScript function. The function to call is encoded the Query String of the endpoint's URL specified in the src-attribute. The name JSONP originates from the fact that JSON encoded data is transformed into JavaScript objects [LKPK15].

Server-side mashups**:** This can be realized with a proxy which integrates data from different endpoints into a common context. The data is then delivered to the clients. With the help of a proxy different origins can be masked and therefore SOP can be circumvented [JaWa07]. For this work, server-side mashups are not applicable because the proxy would know all the information which is only allowed to join a client side [LKPK15].

3.2.1.2.2. **Single-Sign-On**[29]

Complex systems for collaborative research require authentication and authorization. To guarantee a continuous workflow user should not authenticate separately on multiple systems. Therefore the combination of web mashup with a Single-Sign-On mechanism is mandatory. Design decisions regarding authentication and authorization are almost the same:

(1) Implementation in a dedicated component within the system.
(2) Implementation of authentication and authorization mechanisms for each component involved.

In this section we provide an overview of these design dimensions and present several techniques that can be used to implement the various aspects involved.

Non-delegated authentication**:** Here, authentication is handled by each component. One way to implement this is to simply encode user credentials in every request to a component. When having stateful communication, the sessions must be bound to a client.

[29] Cf. "Single-sign-on" in [LKPK15]

For session identification a random ID is commonly generated. The client needs to manage all the active IDs for the different servers in case of stateful communication [LKPK15].

Cookies**:** These are indirectly related to authentication and authorization because cookies are a common way for persisting user sessions in a given request context. The session ID is stored to a local file at client side which represents the cookie. It is transmitted to the host for every request. One cookie cannot be sent to various endpoints in different domains because the SOP also applies to cookies. Therefore cookies cannot be utilized to realize cross-domain Single-Sign-On. In order to persist sessions at the endpoints, cookies can be used in combination with SSO techniques [LKPK15].

***Server-to-server communication*[30]**:** Single-Sign-On can also be implemented with server-to-server communication. Here, opening a session at one endpoint transparently creates sessions on the other endpoints as well by implementing a communication mechanism between servers. As a result, it (a) can be ensured that a single user session is identified by the same token on different endpoints, and, (b) there is no need to send the user's credentials to the endpoints with every request. This technique can be implemented, e.g., with a multicast protocol such as JGroups [00f], sockets or a shared file system. Such an approach is difficult to integrate into Enterprise Security Architectures.

***Access tokens*[31]**:** Typically, SSO solutions are implemented with cryptographic access tokens. Basically, a token is an object that encapsulates the identity and potentially roles of a user as well as a session ID. Tokens generated by one system can be used to perform operations on another system. A token can (and must) be validated by the target system. From a conceptual perspective, using access tokens is not different from the server-to-server communication approach. The only difference is that server-to-server communication is *indirect*, i.e., performed via the client. This makes this approach feasible for implementing SSO between several isolated services on the World Wide Web. Consequently, the approach is, e.g., implemented by Kerberos [NeKo05] and Shibboleth [13]. Access tokens provide a secure communication channel between servers, meaning

[30] Cf. "Server-to-server communication" in [LKPK15]

[31] Cf. "Access tokens" in [LKPK15]

that the client cannot read or modify the content of a token. This is especially useful in our scenario, because it can be used to fulfil an additional non-functional requirement. When implementing access tokens, the main challenges are (1) transferring tokens from the clients to the server, and, (2) key management.

***Rights and roles*[32]:** The handling of authorization of a user's actions is typically coupled with authentication. As a consequence, the design space is closely related to the design space for Single-Sign-On solutions. Role-based access control (RBAC) is an authorization mechanism in which rights are granted to users depending on their associated roles. A role encapsulates a set of permissions. Analogously to SSO, RBAC can be realized with a) a centralized component that authorizes users as well as b) a decentralized solution where every system implements a RBAC component and manages authorization by itself. Important standards for authorization in distributed environments include SAML and XACML [AnLo04].

3.2.1.2.3. **Secure Server-To-Server Communication[33]**

For synchronizing temporary pseudonyms between backend services, secure communication channels are needed. In this context, secure means that the contents of messages are hidden from the clients. This can be implemented with two different mechanisms. Firstly, backend servers can manage exclusive communication channels between them and use these to synchronize information about temporary pseudonyms. Secondly, a secure channel between servers can be built that is routed through the client by using cryptographic tokens.

[32] Cf. "Rights and roles" in [LKPK15]

[33] Cf. "Secure server-to-server communication" in [LKPK15]

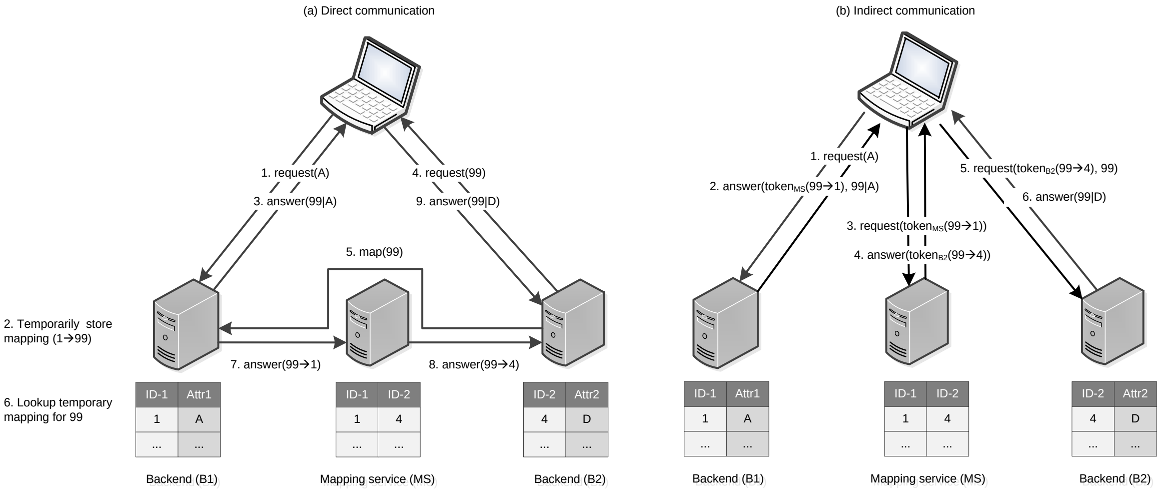


Figure 4. Two basic methods for joining distributed data with temporary identifiers[34]

Direct communication[35]:

Figure 4a shows how the reconstruction of a pseudonymized dataset using temporary identifiers can be performed with direct server-to-server communication. In step 1, the client requests a data item (A) from backend B1. The backend creates a temporary pseudonym for the data entry and persists its association to the actual identifier from its namespace (step 2). The data entry with substituted identifier is then delivered to the client (step 3). Next, the client requests the data item associated with the temporary identifier (step 4) from backend B2. In step 5, the backend requests a mapping of the temporary identifier from backend B1. B1 resolves this request by looking into its set of persisted temporary mappings (step 6). The answer to B2 must be routed through the mapping service (steps 7 and 8). Finally, in step 9, B2 delivers the data entry to the client. Problems with this approach include that (a) it is unclear when exactly the persisted substitution of an identifier may be deleted without implementing complex protocols for transactional guarantees, and, (b) at least seven messages must be exchanged to recombine data distributed amongst two databases via mapping service.

Indirect communication[36]:

[34] Cf. [LKPK15]
[35] Cf. "Direct communication" in [LKPK15]
[36] Cf. "Indirect communication" in [LKPK15]

Figure 4b shows the reconstruction of a pseudonymized dataset with indirect server-to-server communication. Analogously to the previous example, the client requests a data item from backend B1 (step 1). In step 2, the backend creates an association with a temporary identifier, replaces the actual identifier for the data item and sends it back to the client. In contrast to the previous scenario, where the mapping from the actual identifier to the temporary pseudonym is persisted, B1 also sends an encrypted token containing the association. The client forwards the token to the mapping service (step 3) where it is decrypted and the ID from backend B1 is translated in the associated ID at backend B2. Next, the mapping service generates a second token for B2, containing the mapping from the temporary pseudonym to the original identifier. This token is sent to the client (step 4) where it is forwarded to backend B2 (step 5). Finally, in step 6, B2 decrypts the token, performs a lookup for the data item and sends the result back to the client, along with the temporary pseudonym. At the client side, the data from both backends can be joined using the temporary identifier. Compared to direct server-to-server communication, the number of exchanged messages is reduced. Fewer communication channels must be managed, because the same communication channels are used for client-to-server and server-to-server communication. Moreover, as already noted above, indirect server-to-server communication can also be implemented relatively easily, if access tokens are already used for implementing Single-Sign-On. In the remainder of this section, we will elaborate on ways to implement cryptographic (access) tokens.

***Transferring Tokens*[37]**: While tokens can easily be sent from servers to clients in our context, sending tokens from clients to servers is more challenging. Tokens can be embedded into three different segments of an HTTP request: (1) HTTP request line (URL), (2) HTTP header fields, (3) HTTP message body. These techniques have different properties in terms of compatibility to legacy browsers, implementation complexity and compatibility with other techniques required to implement pseudonymized data management, especially JSONP. Here, all parameters must be encoded into a request URL, because, by specification, requests embedded into src-attributes are executed as HTTP GET-Requests by browsers. The only approach that ensures backwards compatibility and that is compatible with JSONP is embedding tokens into the HTTP request line via *URL Rewriting*. An additional challenge when implementing

[37] Cf. “Transferring tokens” in [LKPK15]

this method is to overcome a length restriction that is enforced to URLs by most browsers (e.g., 2083 characters per URL in Internet Explorer). This can be solved by implementing packet fragmentation mechanisms, i.e., splitting a request into multiple sub-requests at the client-side, which are recombined into one request at the server-side [Ietf81]. Tokens can also be embedded into the *header* of HTTP GET- and POST-Requests but this method can only be realized with JavaScript calls and is thus not compatible with JSONP. Finally, tokens can be embedded into the *HTTP Body* of POST-Requests. Again, this is not compatible with JSONP, because JSONP requires GET-Requests to be performed.

***Key management*[38]:** To maintain confidentiality for the contents of a token, symmetric or asymmetric (or hybrid) cryptography can be employed. Depending on the topology of the infrastructure (hierarchical or peer-to-peer), encryption also affects key management. In a hierarchical infrastructure, a single component can be employed to manage all keys needed for the encryption of tokens, whereas in a peer-to-peer infrastructure each component needs to manage key pairs for every other component. In web-based applications, tokens can be implemented with JSON Web Tokens (JWT) utilizing related technologies such as JSON Web Encryption (JWE), JSON Web Signatures (JWS) and JSON Web Keys (JWK) [Jone11].

3.2.2 Evaluation[39]

Considering maintainability (*BCR-7*), our aim was to develop a solution that is robust while only relying on technologies supported by common web browsers (including wide-spread legacy browsers). We therefore decided to build a client-side Web Mashup with HTML Frames for parts of the application that do not require any interface-to-interface communication and with utilization of JSONP for all other cases. To support data collection, as defined by *DER-1*, we realized interfaces for CRUD operations on this tree with two functional views: “*create, list & delete”* and “*view & update”***.** The former provides a list of documents and allows creating new or deleting existing documents. The latter shows the content of a document and allows updating it. Several instances of these two types of views may be displayed next to each other, thus providing an integrated interface as required by *DMR-10* to fulfill our functional requirements DMR-7, DER-3 and *DER-4*. JSONP is a good

[38] Cf. “Key management” in [LKPK15]
[39] Cf. “Technical design” in [LKPK15]

solution for interfaces in which data of many entities has to be displayed, i.e. the "*create, list & delete"* view. In the other cases, i.e. the *"view & update"* interface, we leverage HTML Frames because of their ease of implementation and therefore increased productivity when developing the software.

For Single-Sign-On (see *DMR-11*) our solution implements non-delegated authentication where each component handles authentication and authorization autonomously. This design decision is driven by the fact that many pseudonymization schemes require at least one trusted third party (TTP), which is organizationally and physically separated from the rest of the system. As a result, decentralized authentication and authorization is performed for each request and each component provides its own administrative interface and RBAC model. Keys are distributed in a peer-to-peer topology. Non-delegated authentication is implemented with cryptographic access tokens that also provide a secure communication channel between servers that is routed via the client. Tokens are created by backend servers. At the client side, they are always appended to the URL. When using JSONP this is implemented with JavaScript, otherwise a server-side URL rewriting mechanism is used. Session-IDs are persisted with cookies. As defined in requirement *LSR-22*, our solution supports two-tier pseudonymization. This makes joining distributed data more complex, because pseudonyms need to be translated from one namespace into another namespace. Moreover, requirement *LSR-20* specifies that this linkage can only to be performed at the clients. We use the same token infrastructure for SSO and for implementing an indirect communication channel between backend servers (cf. Section 3.3.2.3 and Figure 4b). There are multiple frameworks for implementing token infrastructures, but we decided to develop our own solution that is tailored to our requirements for the following reasons. JSON Web Tokens were at the time of implementation still in a draft-phase and therefore immature. XACML and SAML come with a significant overhead regarding the size of the exchanged messages because they use an XML-Syntax. This is problematic when transmitting data via URLs. Furthermore, XACML and SAML are complex, resulting in a rather high implementation effort. In our system, tokens are encrypted with a hybrid method combining AES and RSA. The payload is encrypted symmetrically and integrity protected and the key for decryption K_1 is encrypted asymmetrically with the public key K_2 of the receiver. Tokens contain the key *K1, username, password, counter* and *payload data P* (e.g. encoded in JSON syntax): $E_{K2}(K_1)$, $E_{K1}(username, password, counter, P)$. Replay protection is implemented with a counter that is continuously incremented and prevents repeated acceptance of tokens by any receiver.

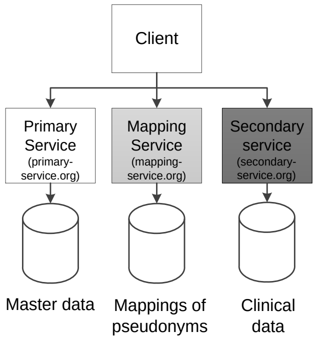

Client
Primary Service (primary-service.org)
Mapping Service (mapping-service.org)
Secondary service (secondary-service.org)
Master data
Mappings of pseudonyms
Clinical data

Figure 5. System architecture for a tightly coupled solution with pseudonymization[40]

The design of our solution supports two or more physically distributed data stores (*PSR-9*) and one or more mapping services (*LSR-22*). All endpoints have to provide API access and all services but the mapping service must be able to provide HTML-formatted data to clients as well. However, the mapping service must provide HTML-Frames that embed HTML-formatted data from other services, as will be explained below. A basic design fulfilling all requirements of the model by Pommerening et al. [PDHG14] must implement separation of master data and clinical data [Fede09]. A minimal solution is shown in Figure 5. The central component is implemented by the backend managing master data (primary service), because it stores the root nodes of the tree and is thus the starting point for user interactions.

Our final solution combines the above techniques into a Web-Mashup that integrates pseudonymized data (*LSR-20*). The first variant, which uses HTML-Framesets, is sketched in Figure 6. Here, a static frame at the top displays selected data of a single entity from the primary service. The content of the second frame, which is located at the bottom, is provided by the mapping service and contains an additional nested frame (nested frame), which shows the corresponding clinical data. Please note that in Figure 6 pseudonyms are represented as clear text instead of being encoded into tokens for the sake of readability. In our solution pseudonyms are encoded into encrypted tokens and therefore never visible to the client (*LSR-21*).

[40] Cf. [LKPK15]

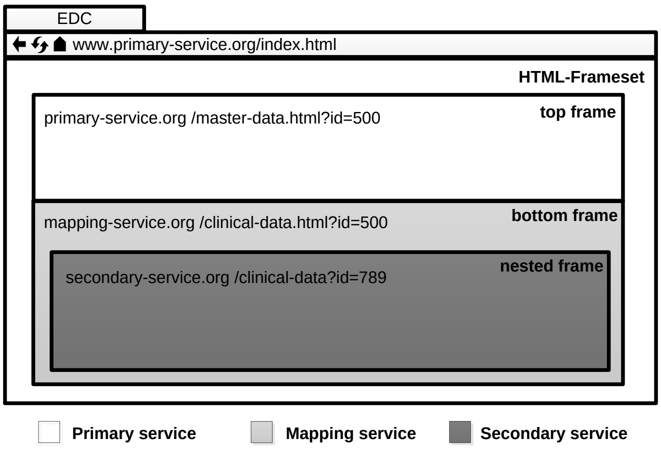

EDC
www.primary-service.org/index.html
HTML-Frameset
primary-service.org /master-data.html?id=500
top frame
mapping-service.org /clinical-data.html?id=500
bottom frame
secondary-service.org /clinical-data?id=789
nested frame
Primary service
Mapping service
Secondary service

Figure 6. Mashup with nested HTML frames[41]

The design of a mashup using HTML-Framesets or IFrames is depicted in Figure 7a. A frameset comprising a top-frame where data from data-store1 displayed and a second frame displays data from data-store2. Due to SOP, the contents of top-frame will be loaded separately and isolated from the contents of frame-2. Only the top frame is able to control the content displayed in frame-2. An interaction of frame-2 with top-frame is not possible. In Figure 7b data from two different domains is displayed together using an IFrame. The IFrame, displaying data from data-store2, is nested in the body of the HTML-document from data-store1. Analogously, to the HTML-frameset the SOP applies in this case as well. The IFrame content is not able to interact with the surrounding contents of the other domain.

[41] Cf. [LKPK15]

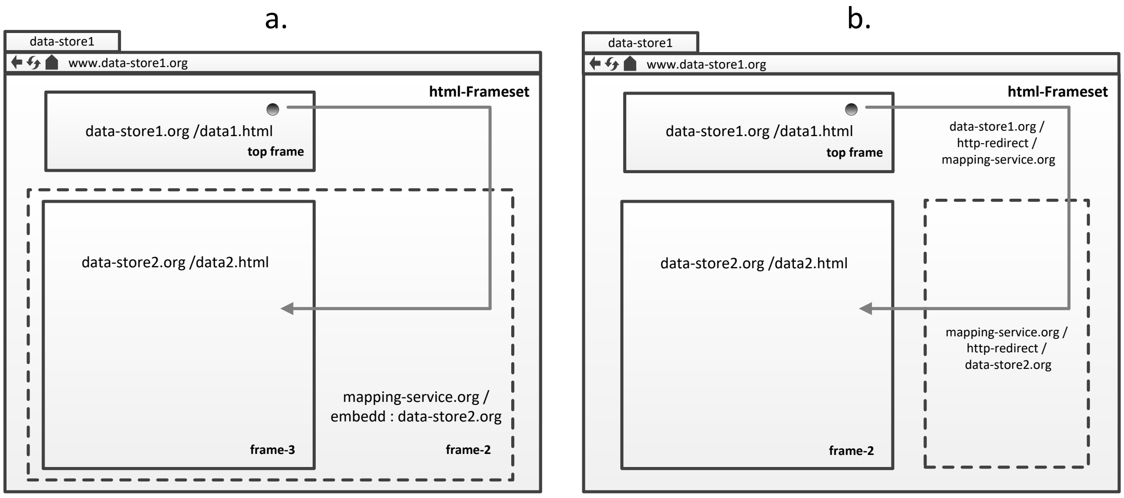


Figure 7. Mashup with HTML-framesets and nested HTML frames or HTTP redirects[42]

A typical workflow in which the above method is utilized is the creation of a new eCRF. Firstly, the user logs into the primary service and selects a specific subject. The primary service returns a HTML-Frameset as response, where the top-frame contains an HTML-document with the master data of the selected subject. A new instance of a predefined eCRF is generated and the resulting document is displayed using the previously described method. In this process, a chain of HTTP-Requests is generated, in which the user's credentials are encoded into tokens and distributed to all endpoints to implement SSO.

[42] Cf. [PKLK15]

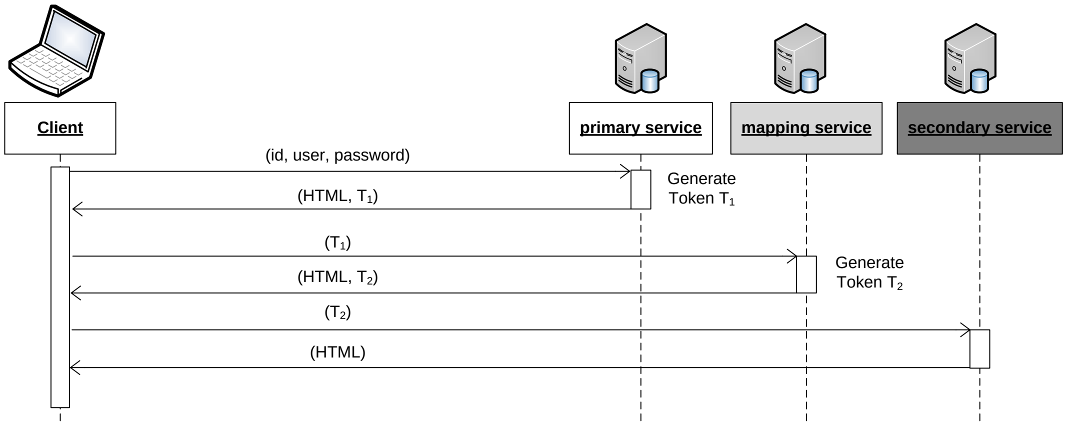


Figure 8. Information flow between services[43]

A basic version of this process is shown in Figure 8. To simplify our illustration, we assume that the first request, which also logs the user into the system, already contains the ID of the subject for which a new document is to be created. In a real-world scenario, the login process would already have been performed earlier. It can be seen that the user's credentials and the id of the data element that is to be displayed are sent to the primary service with the first request. From there on, the operation to be performed, on which data it is to be performed and for which user, is encoded into tokens. These tokens are generated at the backends. This also provides a transparent SSO mechanism. As an alternative to embedding nested frames, this process can also be implemented by using an HTTP-Redirect to route the request from the mapping service to the secondary service.

The second variant of our Web-Mashup uses JSONP requests to display distributed pseudonymized data. It is especially suitable for scenarios in which a larger set of distributed but related entities, e.g., a list of all subjects and an overview of associated clinical data, is to be displayed. The method is sketched in Figure 9 and Figure 10.

[43] Cf. [PKLK15]

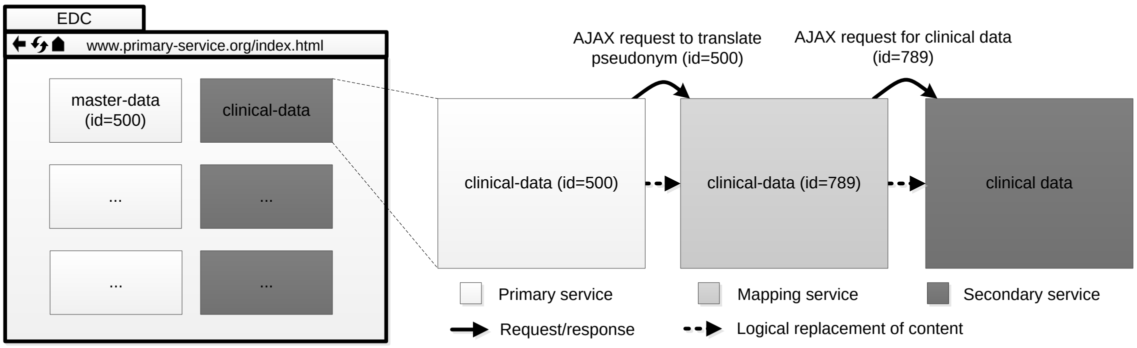


Figure 9. Workflow and assembly (i) of pseudonymized data using JSONP[44]

First, a HTML-document is delivered to the client by the primary service, e.g. containing the master data of multiple subjects as well as a set of pseudonyms of related data items for each subject.

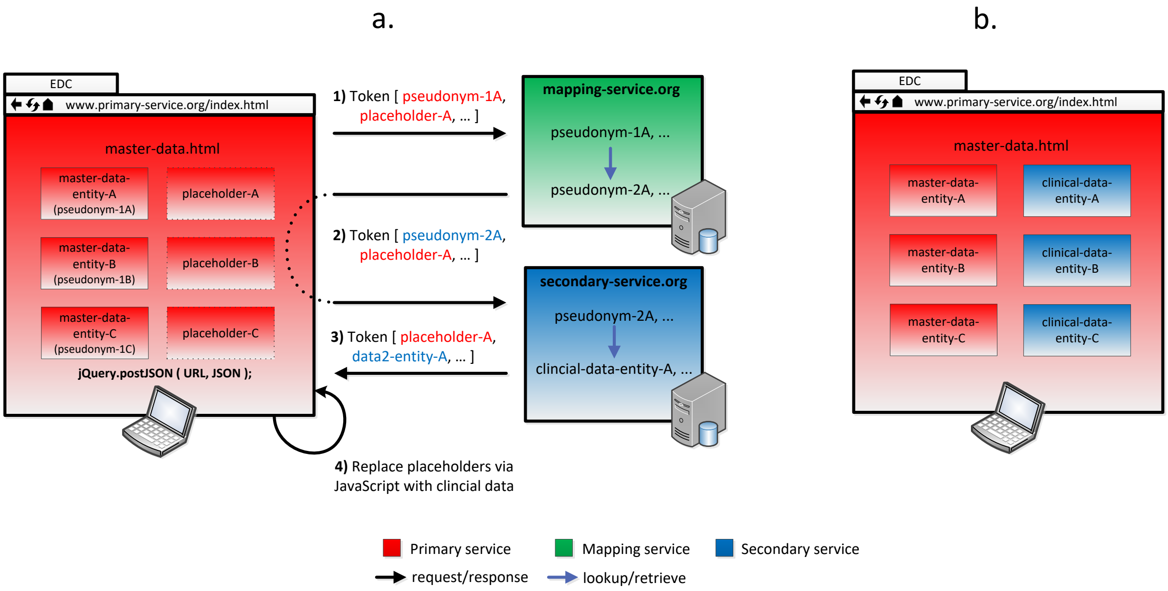

a.
b.
EDC
www.primary-service.org/index.html
master-data.html
master-data-entity-A (pseudonym-1A)
placeholder-A
master-data-entity-B (pseudonym-1B)
placeholder-B
master-data-entity-C (pseudonym-1C)
placeholder-C
jQuery.postJSON (URL, JSON);
1) Token [pseudonym-1A, placeholder-A, ...]
2) Token [pseudonym-2A, placeholder-A, ...]
3) Token [placeholder-A, data2-entity-A, ...]
4) Replace placeholders via JavaScript with clincial data
mapping-service.org
pseudonym-1A, ...
pseudonym-2A, ...
secondary-service.org
pseudonym-2A, ...
clincial-data-entity-A, ...
EDC
www.primary-service.org/index.html
master-data.html
master-data-entity-A
clinical-data-entity-A
master-data-entity-B
clinical-data-entity-B
master-data-entity-C
clinical-data-entity-C
Primary service
Mapping service
Secondary service
request/response
lookup/retrieve

Figure 10. Workflow and assembly (ii) of pseudonymized data using JSONP[45]

Via JavaScript code, the client then performs a set of AJAX requests to translate the pseudonyms from the primary service's namespace to the namespace of the secondary service. Finally, the data items identified by those pseudonyms are requested from the secondary service and the content of the HTML document is updated dynamically. The

[44] Cf. [LKPK15]
[45] Cf. [PKLK15]

described process is shown in Figure 10a and its result is depicted in Figure 10b. The basic information flow is very similar to the one which is depicted for the method implemented with HTML Frames in Figure 7. The only difference is that the primary service and mapping service only return tokens, which are then passed to the next receiver, thus implementing the previously described communication channel that is routed via the client.

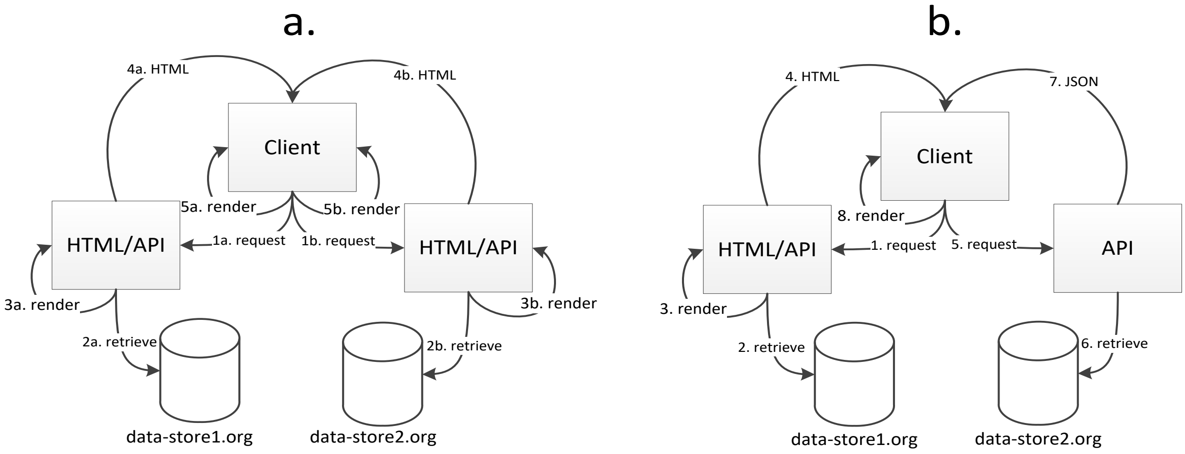


Figure 11. Basic information flow for mashups using HTML-framesets or JSONP[46]

Figure 11 illustrates information flow and structure of the system depending on the techniques utilized (HTML-framesets Figure 11a or JSONP Figure 11b). The client sends a request to data-store1 and receives a HTML page which contains a pointer to data-store2. This pointer invokes another request to data-store2 where data is retrieved and sent back in HTML format.

3.2.2.1 Security Concept

The design of our security concept was driven by adherence to European General Data Protection Regulation [Jour16] and the German Data Protection Act for Data Security in Research Networks [Fede09] and compliance with the well accepted concept of TMF [PDHG14]. Our concept is based on three principles: “Keep the outsiders out”, “Principle of least privilege” and “Defense in depth”. It has a multi-layer security model with spatial and

[46] Cf. [PKLK15]

organizational separation of identification data, clinical data, specimen and image related data.

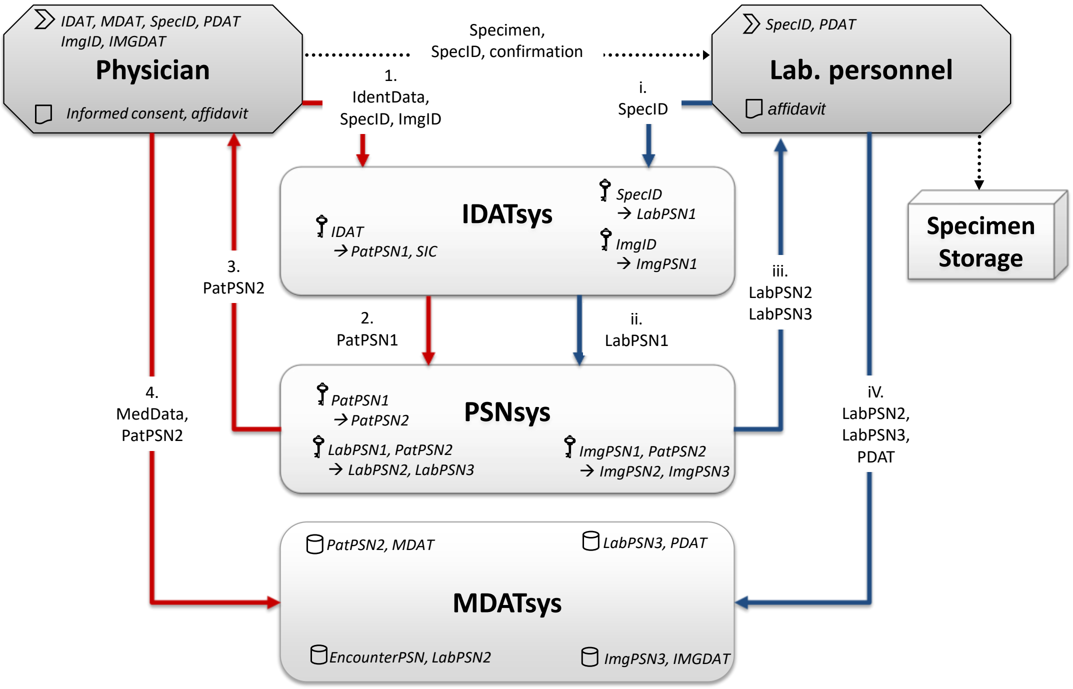


Figure 12. Security concept overview[47]

We divided the collected data into different types according to [PDHG14]. Identification data, *IDAT* (e.g. Name, address, contact information, etc.), medical or clinical phenotype data, *MDAT* (e.g. diagnosis, symptoms, weight, etc.). Data resulting from the collection and management of biospecimen (*PDAT*) as well as data related to imaging (*IMGDAT*). These data have to be pseudonymized and separated (*PSR-9, PSR-10*). Therefore separate subsystems (*IDATsys*, *PSNsys*, *MDATsys*) are introduced which store and manage the different types of data. The concept and workflow are illustrated in Figure 12 and was published in a similar version in [Fkoh10].

The Identification Service subsystem (*IDATsys*) stores identification data of all patients. Furthermore, it manages all visible identifiers of patients, biospecimen and images and

[47] Cf. [Fkoh10]

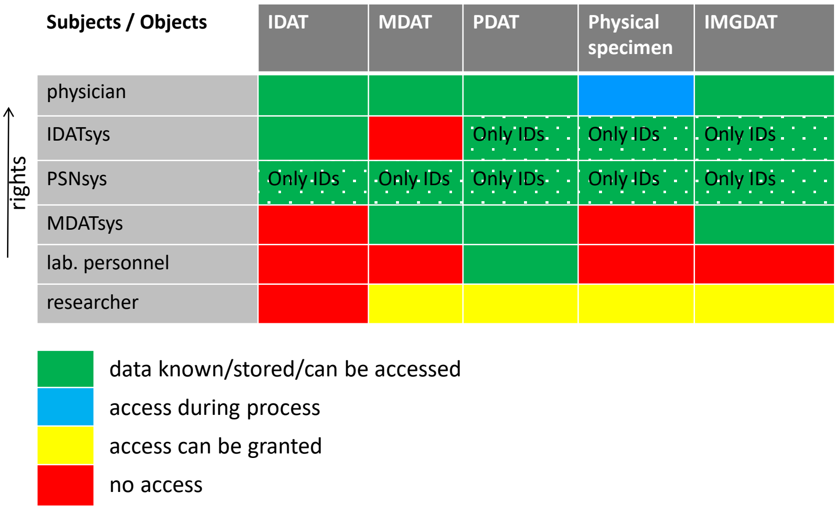
maps them to first-level, system internal pseudonyms that are invisible. The term ‘invisible’ indicates that such a pseudonym will never be revealed to a user or application administrator (*LSR-21*). All internal pseudonyms are unique random alphanumeric strings. Furthermore, *IDATsys* also generate visible identifiers which are random strings as well. An example is a study identification code (*SIC*). The *SIC* serves as a visible identifier for filing at site and to establish a link to a patient’s electronic health record (*EHR*). In addition, the *SIC* can be utilized for communication, e.g., when a patient has agreed to move from one site to another (*DMR-12*). Here, administrators use the *SIC* to assign the registry record of the patient to a new site according to established SOPs. Due to encryption, the identification data of a newly entered patient cannot be compared with existing entries of all sites, e.g., for the detection of duplicates. Possible duplicate patient entries are indicated to a user by the *IDATsys* (*DMR-12*). This is performed by automatically generating pattern via phonetic algorithm and hashing using the entered identification data. These patterns are stored for each patient and can be compared with one another. If a match is found, the system displays to a user that a similar entry exists and indicates to ask the patient if recruitment has already been performed by another site. A found match is afflicted with certain likelihood, because there may be different persons with similar identification data. In case of a match that is confirmed by a patient and consent has been granted to merge the records, identification data records can be merged by an administrator according to SOPs.

The basic task of the Pseudonymization Service (*PSNsys*) is the mapping of invisible system internal pseudonyms to one another (*LSR-22*). *PSNsys* maps internal first-level pseudonyms for identification- , biospecimen and image related data that were generated by *IDATsys* to second-level pseudonyms, generated by *PSNsys*. The pseudonyms for biospecimen and images subdivide into second-level pseudonyms to establish a link to a patient’s encounter in the clinical data (*MDAT*) and second-level pseudonyms for the image- or biospecimen related data itself (*PDAT*, *IMGDAT*). The second-level pseudonyms for *PDAT* and *IMGDAT* are utilized to create an indirection between the link of biospecimen and images to an encounter. Hereby, it is possible to separate the information of biospecimen- or image management data (e.g. identifiers, dates) and clinical data in general. By mapping only pseudonyms, the *PSNsys* is not aware of any kind of other data but crucial as linking component for data of all other subsystems. *PSNsys* realizes a two-tier pseudonymization between identification- and clinical data, between specimen/image- and identification data as well as between specimen/image- and clinical data (*LSR-22*). It distributes the second-level pseudonyms to *MDATsys* where they are assigned to clinical- and biospecimen- or image related data.

The subsystem *MDATsys* stores second-level patient pseudonyms and all clinical parameters as well as the mapping of second-level biospecimen- or image pseudonyms to a patient's encounter. Furthermore, *MDATsys* stores image- and biospecimen related data (*PDAT*, *IMGDAT*) with the corresponding second-level pseudonyms.

Rights and Roles

When implementing a pseudonymization architecture, the permission model for access to data has to be designed carefully (e.g. following the need-to-know principle as well as the principle of least privilege) (*LSR-13, LSR-14, LSR-19*). Audit trails are essential in any case (*DBR-2*). Context-dependent rights and roles of users are an important factor: who (in which role) has the right and the need to know which data in which context.



Subjects / Objects	IDAT	MDAT	PDAT	Physical specimen	IMGDAT
physician	🟩	🟩	🟩	🟦	🟩
IDATsys	🟩	🟥	🟩 Only IDs	🟩 Only IDs	🟩 Only IDs
PSNsys	🟩 Only IDs	🟩 Only IDs	🟩 Only IDs	🟩 Only IDs	🟩 Only IDs
MDATsys	🟥	🟩	🟩	🟥	🟩
lab. personnel	🟥	🟥	🟩	🟥	🟥
researcher	🟥	🟨	🟨	🟨	🟨

🟩	data known/stored/can be accessed
🟦	access during process
🟨	access can be granted
🟥	no access

Figure 13. Access rights of subsystems and actors[48]

In general, a health care professional treating a patient will have more permissions than a researcher (*LSR-23*). User-roles comprise application-administrators, monitors, physicians and lab personnel. Each role has different permissions which are create-, read-, update-

[48] Cf. [PKLK15]

and delete-operations (*CRUD*). These transactions are executed distributed (*DMR-9*). An overview of permissions in the system can be seen in Figure 13. *CRUD* Permissions of user accounts is granted to application-administrators. These administrators do not have access to any patient data, neither identification nor clinical data (*LSR-23*). For review purposes data manager permissions cover read-operations for clinical data of patients. For physicians and their team *CRUD* permissions are enabled for patient, biospecimen and image related data (of affiliated site). Data managers can access clinical data from all patients without having access to any identification data. In case inconsistencies or missing information was found, data managers can add queries to a patient's encounter (*DMR-6*). This result will be displayed to physicians or staff. After the physicians have revised entries, data managers can lock single encounters which will set the clinical information to read-only and no further modifications are possible (*DMR-7*). Data managers can unlock an entry at any time, if necessary. In case of questions of a data manager concerning single subjects that need to be communicated personally via telephone between physician and data manager, the study identification code (*SIC*) to retrieve the entry (*DMR-10, LSR-18*).

Permissions for lab personnel cover create-, read- and update-operations for biospecimen in the system. Lab personnel do not have access to any identification or clinical data of patients in the system. For QA-purposes lab personnel can review biospecimen entries made by physicians and cross-check the information with the biospecimen documentation sheet which is sent by physicians to the lab along with the biospecimen. In case of inconsistencies or missing information, lab personnel can add a query to the biospecimen entry in the system (*DMR-6*). This query will be presented to the physician after login. If biospecimen arrive at the lab, without having already been entered into the system by physicians, lab personnel can create an orphan entry in the system. Here all information of the accompanying documentation sheet is entered. The physician or staff can afterwards assign the entry created by the lab to the corresponding patient encounter using the biospecimen identifier. The entry can always be revised by the physician or staff.

For researchers, a formal process of data release has been established. Researchers formulate their request in a Data Use Agreement (*DUA*) which is checked for adherence to the patients' informed consent (*ER-4, ER-5, ER-6*). A *DUA* describes the exact purpose and the research question, and the signers confirm data will not be misused (*DMR-14*, *LSR-18*). Next, the request undergoes a review process, led by a scientific steering committee (*ER-4*). Requests of researchers for data export and access to samples and to results of corresponding analyses always have to adhere to the patients' consent (*DMR-14*). After approval of the steering committee, the *DUA* is signed by all data controllers (*ER-4*). Exported data are annotated with a random pseudonym which is generated for each exported dataset. The exported data are in conformance with HIPAA limited data set [Usde13]. Furthermore, exported data are encrypted with for secure handover to researchers. Internal identifiers and pseudonyms are never exported (*LSR-21*).

3.2.2.2 System Instances[49]

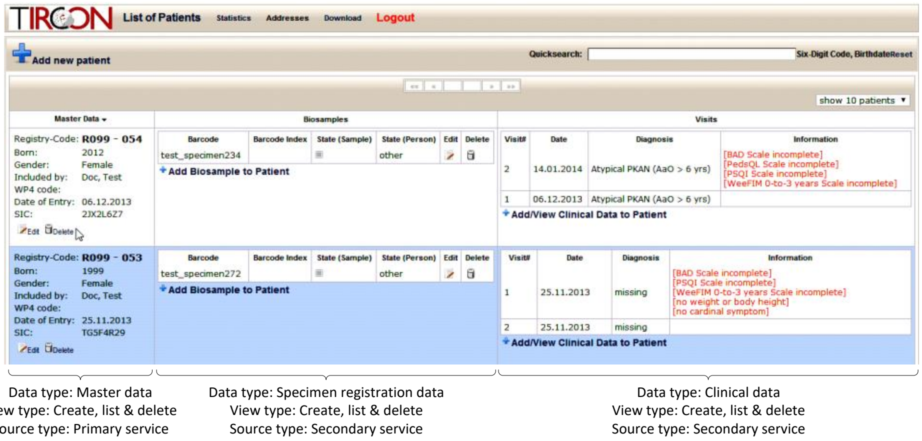
We have used the described solution as a basis for implementing the data management software for two large research networks for rare diseases. The primary actors are health care professionals in an observational study. No specific intervention takes place, and data used for research are collected during health care activities. We will not further describe the associated biobanks, which use prepared "kits" (tubes with identifiers sent to sites and returned to a central biobank) with pseudonymous labels. Our first system instance is being used in the mitoNET project [BGLK12]. MitoNET is a research network for mitochondrial disorders which was started in 2009 and has been funded by the German Federal Ministry of Education and Research (BMBF). It serves as a platform for over 18 centers in Germany and is currently in its second funding period. By February 2015 about 1100 patients have been recruited. Data is managed by 35 eCRFs, which comprise over 900 attributes. Our second system instance also supports a research network for neurodegenerative diseases, TIRCON [KLKB12]. This project was started in 2012 and is funded by European Commission FP7-Health Work Programme [Euro12]. Here, our software provides an integrated EDC system for 13 partners from 8 countries (including the US, UK and Germany). Institutional Review Boards (IRBs) and data protection officers of the participating sites have approved the concept. By February 2015 about 200 patients have been recruited. Data is collected in 34 eCRFs consisting of almost 1000 attributes. Both systems are compliant with the German pseudonymization concept by Pommerening et al. [PDHG14].

In these projects the following separated and two-tier pseudonymized data pools are managed by our solution: a) master data, b) clinical phenotype data and c) biospecimen registration data.

[49] Cf. "Implementations" in [LKPK15]

In both projects our solution was implemented with Java-Server-Faces as the driving technology for the backends, jQuery for client-side functionalities, MySQL as a database system as well as Tomcat application servers and Apache web servers as runtime environments. Both systems use two-factor authentication with One-Time-Passwords (OTP) following the OATH standard [00g] for user accounts with high privileges. Users are provided with time-based dongles that generate short-living passwords, which can be used to access the system exactly once. Communication between endpoints and the clients is secured with Transport Layer Security (TLS/SSL). Automated penetration-tests have been performed to detect weaknesses. Master data is stored encrypted in the according backend. Accountability and integrity are ensured by an audit-trail that keeps protocol of every data modification on each backend. We use virtual servers to provide fail-over mechanisms. All endpoints are secured by firewalls. Encrypted backups are created daily and transferred to one dedicated location per backend. Both systems were designed and implemented at our institution in close collaboration with the involved physicians and researchers using an agile development process with short feedback cycles.

Both systems provide web-based data entry, support of cross-validation and plausibility checks, a (logical) central database, and an elaborated security concept with multi-tier pseudonymity for patient-, specimen- and image-identifiers. A web-browser is the only software needed to access the system. The informed consent serves as basic agreement for the patient's research participation. The systems use controlled vocabularies [RKBS08] as well as standardized questionnaires [BaVA99, ScPE12, SMBB06]. Access roles comprise application administrators, monitors, physicians and lab personnel. Each role has different permissions in terms of create-, read-, update- and delete operations (CRUD) for certain types of documents and system objects. Application administrators are able to perform all CRUD-operations on user accounts but do not have access to any type of research data. Monitors may perform read-only operations on clinical data to perform quality assurance. Physicians may perform all CRUD-operations on master data and clinical data. Each physician and patient is associated to his or her home institution. Physicians are only able to access data from patients related to the same institution.



Registry-Code: R099 - 054
Born: 2012
Gender: Female
Included by: Doc, Test
WP4 code:
Date of Entry: 06.12.2013
SIC: 2JX2L6Z7

Barcode	Barcode Index	State (Sample)	State (Person)	Edit	Delete
test_specimen234			other		

Visit#	Date	Diagnosis	Information
2	14.01.2014	Atypical PKAN (AaO > 6 yrs)	[BAD Scale incomplete] [PedsQL Scale incomplete] [PSQI Scale incomplete] [WeeFIM 0-to-3 years Scale incomplete]
1	06.12.2013	Atypical PKAN (AaO > 6 yrs)	

Registry-Code: R099 - 053
Born: 1999
Gender: Female
Included by: Doc, Test
WP4 code:
Date of Entry: 25.11.2013
SIC: TG5F4R29

Barcode	Barcode Index	State (Sample)	State (Person)	Edit	Delete
test_specimen272			other		

Visit#	Date	Diagnosis	Information
1	25.11.2013	missing	[BAD Scale incomplete] [PSQI Scale incomplete] [WeeFIM 0-to-3 years Scale incomplete] [no weight or body height] [no cardinal symptom]
2	25.11.2013	missing	

Data type: Master data
View type: Create, list & delete
Source type: Primary service

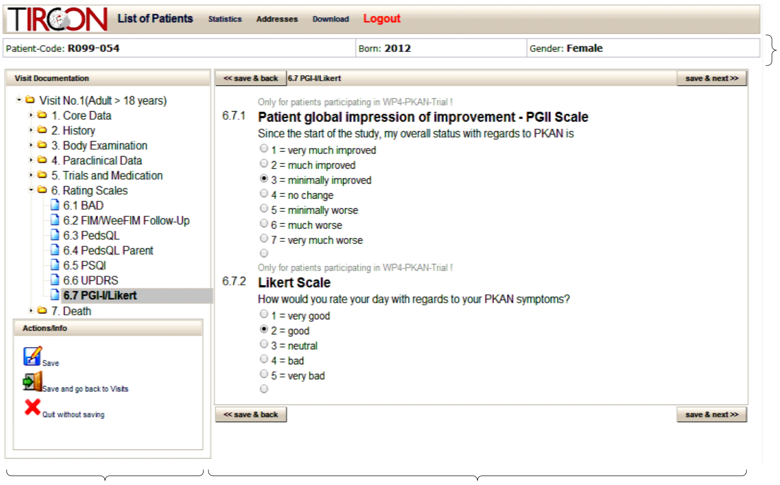
Data type: Specimen registration data
View type: Create, list & delete
Source type: Secondary service

Data type: Clinical data
View type: Create, list & delete
Source type: Secondary service

Figure 14. List of patients (JSONP mashup)[50]

An example screenshot of the EDC system implemented for the TIRCON project is shown in Figure 14. Here, a seamless integration of data from different pools is implemented, providing the “*create, list & delete”* functionality defined previously. The view shows an overview and summary data about all subjects, which can be managed by the current user. For each subject, the list is substructured into master data used for re-identification and an overview of the documents used to track biosamples and to collect clinical data. The view is realized with JSONP.

[50] Cf. [LKPK15]



Patient-Code: **R099-054** | Born: **2012** | Gender: **Female**

- Visit No.1(Adult > 18 years)
  - 1. Core Data
  - 2. History
  - 3. Body Examination
  - 4. Paraclinical Data
  - 5. Trials and Medication
  - 6. Rating Scales
    - 6.1 BAD
    - 6.2 FIM/WeeFIM Follow-Up
    - 6.3 PedsQL
    - 6.4 PedsQL Parent
    - 6.5 PSQI
    - 6.6 UPDRS
    - **6.7 PGI-I/Likert**
  - 7. Death

6.7 PGI-I/Likert

Only for patients participating in WP4-PKAN-Trial !

6.7.1 **Patient global impression of improvement - PGII Scale**

Since the start of the study, my overall status with regards to PKAN is

○ 1 = very much improved
○ 2 = much improved
● 3 = minimally improved
○ 4 = no change
○ 5 = minimally worse
○ 6 = much worse
○ 7 = very much worse
○

Only for patients participating in WP4-PKAN-Trial !

6.7.2 **Likert Scale**

How would you rate your day with regards to your PKAN symptoms?

○ 1 = very good
● 2 = good
○ 3 = neutral
○ 4 = bad
○ 5 = very bad
○

Data type: Master data
View type: View & update
Source type: Primary service

Data type: Clinical data
View type: Create, list & delete
Source type: Secondary service

Data type: Clinical data
View type: View & update
Source type: Secondary service

Figure 15. eCRF for entry of clinical data (Mashup with HTML-Framesets)[51]

A second screenshot from the TIRCON application is presented in Figure 15. It shows an integrated view of master data and clinical data from an eCRF realized with a HTML-Frameset, which is provided by the primary service. The view implements the previously defined functionality of *view & update*. A top-frame displays the master data of a select subject, whereas the bottom-frame shows the associated documents with clinical data, which are stored at the secondary service. The bottom-frame is organized into two interlinked regions. Firstly a document tree provides an overview of the different documents available for the subject. Secondly, the currently selected document from the tree is displayed.

The informed consent serves as basic agreement for the patient's research participation. After consent was granted, data entry is performed at each encounter of a patient. Controlled vocabularies [RKBS08] are used for data entry of clinical-phenotypical data, as well as standardized questionnaires [BaVA99, ScPE12, SMBB06]. Specimen are taken, prepared according to SOPs and stored and managed in a central repository. The collection of specimen is performed using sealed pre-configured kits issued by a central lab. Those kits contain an informed consent, affidavit, documentation sheet and several specimen tubes which are tagged with the kit's identifier. An affidavit, comprising only a pseudonym

[51] Cf. [LKPK15]

and physician’s signature, is sent to the lab to keep record whether the patient’s informed consent was granted and archived at site.

3.3 Discussion

3.3.1 Principal Results[52]

In this work, we have presented a systematic overview of challenges that may be faced and techniques that may be used when implementing pseudonymized data management with web technologies. The presented methods match the use cases requiring an integrated view on the distributed data. Moreover, we have described a solution that can be tailored to different pseudonymization schemes by using a well-defined subset of the presented techniques. Our solution is independent of the actual distribution of data and it is able to manage associations between patients or visits and further external entities. The aim of our solution is to build integrated applications, in which the actual distribution of data is transparent to users, providing a virtual central database. Our solution features single-sign-on, supports multi-tier pseudonymity and does not require direct server-to-server communication. By providing various features, our solution can be used for the collection of a broad spectrum of different types of data in compliance with national and international laws. Moreover, as a basis, we chose a set of techniques that are supported by modern state-of-the-art browsers as well as legacy browsers. We have shown the practical applicability of our approach, by using it as a basis for implementing two geographically large research networks. Both systems have been in productive use for several years. Several national and international Institutional Review Boards (IRBs) and Data Protection Commissioners of the participating sites have approved the concept.

As a result of separating a datasets, relationships between data subsets must be maintained. The properties of the relationships and their management are also relevant to the privacy properties of the overall architecture. Moreover, they determine the ability of an IT system to support specific functionalities.

Central questions to be asked when designing a pseudonymization scheme are:

1. *What are the threats and threats/vulnerabilities relevant to our context?*
2. *How can separation/pseudonymization be used to mitigate them?*

[52] Cf. “Principal Results” in [LKPK15].

3. *What implications in terms of potential threats do different implementations have?*
4. *How can re-identification be performed when having separation be implemented?*
5. *Where do high costs of different implementation variants occur?*

"A key principle of pseudonymization is the separation of contexts" [PKLK15]. Here, our approach covers a broad spectrum of logical, physical, technical and organizational measures.

This definition of *pseudonymization* allows for a broad spectrum of technical and operational measures to be used for its implementation. Currently, *multi-tier pseudonymization* is understood as multiple coding of surrogate identifiers.

"In typical information system architectures of our domain, separation is implemented by distribution. Distribution is realized by combinations of technical and organizational measures. Partitions are deployed on different backend servers, and servers are placed in different locations in different organizational units. On a technical level, this results in different physical access controls and separate hardware, often combined with different logical access controls" [PKLK15].

On an operational level, different persons are responsible for the backend servers, and they report to different supervisors [PSMS08].

In our solution, confidentiality of identification data in case of server maintenance procedures is ensured by encryption in IDATsys. Here, server administrators can only see encrypted patient data without being able to access the key for decryption. The informed consent with directly identifying data (e.g. name, signature) remains at each recruiting site. As a result, the lab does not learn any identification data. This solved a general problem concerning data privacy of the postal transport of specimen. If problems or questions with patient entries arise (e.g. during monitoring by a data manager) that need to be discussed between monitor and physician, the *Study Identification Code (SIC*) is utilized as identifier for a patient. The *SIC* is a random alphanumeric pseudonym which is searchable in the system for physicians or monitors to find the corresponding entry. This helps with avoiding that identification data is communicated outside the system (e.g. name or birthdate of patients).

The design of symmetric integration of identification data, clinical data, biospecimen- and image related data allows other data sources to be integrated in the same manner. This can be important in case of re-use of clinical data. Decentralized labs, analogous to the central lab presented here can easily be integrated in the architecture. There is no need of re-labeling specimen identifiers during collection and management, because the specimen data is stored with the invisible system internal pseudonym and without link to clinical information. If biospecimen are handed out to researchers according to *Data Use Agreements* (DUAs), biospecimen are re-labeled with additional random identifiers, generated during export process.

3.3.2 Comparison with Prior Work[53]

The work presented is not the first solution that has been proposed for pseudonymized data management. It is one of the very few contributions, however, asking fundamental questions. We have presented a systematic solution for a typical use case, but we strongly suggest further work. Moreover, we have put emphasis on detailed descriptions of alternatives that are available for implementing the methods described here.

Already on the concept level, there is a variety of different approaches to pseudonymization. We refer to the overview presented by Aamot et al [AKRK13]. In this work, we will focus on two models: the ISO Technical Specification 25237 [Inte00] on “Health informatics - Pseudonymization”, which describes concepts fundamental to pseudonymity in biomedical research environments, and the German model by Pommerening et al. [PDHG14], which is closely related to ISO 25237. For [PDHG14], there is no comprehensive description in English, so we can only refer to overviews presented in [HDRP10, PSMS08]. The articles by Brinkmann et al. [BKGÜ05], Spitzer et al. [SpUU09] and Lablans et al. [LaBÜ15] are also based on the German model and contain short summaries.

When creating a mashup, there are various approaches to overcome the limitations of SOP. The work by Crites et al. [CrHC08] proposes an object abstraction model that utilize public interfaces for websites from different domains to communicate with each other. De Keukelaere et al. present in [KBSC08] an approach that encapsulates contents from trusted domains in components which are loaded in an IFrames. These IFrames again contain

[53] Cf. “Comparison with related work” in [LKPK15].

invisible “tunnel” IFrames with are employed to communicate with the main application. In the paper by Jackson and Wang [JaWa07] a proxy is used to mask the different origins of the content to the client. The communication logic in [JaWa07], [CrHC08] and [KBSC08] is provided by a custom JavaScript libraries.

Several articles have described systems that implement loose coupling. For an in-depth comparison of loosely coupled and tightly coupled architectures we refer to Section “Architectural Options”, but we feel that the most important drawback is that users may need to manually transfer pseudonyms between component systems. The work by Eggert et al. uses a paper-based core process in which pseudonyms are printed on documents [EWAG07]. Physicians use the pseudonym from the paper-based documents for remote entry of clinical data. Moreover, a trusted third party is involved in the re-identification process. Demiroglu et al. have published two articles describing loosely coupled systems that implement one-tier pseudonymity [DeSS11, DSQS12]. Both systems manage links to two external systems: Starlims [00h], which is used for managing biospecimen and secuTrial [00i], which acts as a clinical phenotype database.

The most elaborated approach for loose coupling has been presented by Lablans et al. in [LaBÜ15]. Their work describes a reference implementation of a REST-based interface for the realization of clinical research networks. Its main functionality is to support identity management, i.e., to store master data together with an associated pseudonymized link (e.g. identifier) to an external data pool. In the article, the EDC system secuTrial [00i] is used as an example. Analogously to our approach, the authors utilize tokens for the communication between the clients and the RESTful backend but these tokens are not cryptographically protected. Other differences to our solution include that their system only supports one-tier pseudonymization and that internal pseudonyms used for storage are visible to users.

The work by Brinkmann et al. [BKGÜ05] is an implementation of the model by Pommerening et al. The system provides an integrated view on data from two separated pools in a web browser. It employs IFrames for a tight coupling of one-tier pseudonymized master data and DICOM images. Temporary identifiers are utilized to integrate these data without making pseudonyms visible to clients. While these design decisions and implementation methods are similar to our solution, it is much narrower in its scope. The system focusses on associating a collection of images with patient master data, which is a rather simple setup with a data model that is not too complex. The system only uses Iframes for providing an integrated view on distributed data. IFrames are well suited for simple data structures, but have technological limitations when dealing with more complex data. We have also verified this with experiments. It can therefore be assumed that the system is not able to efficiently provide comprehensive views on complex structures consisting of multiple different entities that are interlinked with high multiplicities. Moreover, the system provides a smaller set of features than ours. Important examples include not supporting multi-tier pseudonymity and not providing alternatives to direct server-to-server communication for the synchronization of temporary identifiers.

In [SpUU09] Spitzer et al. extend this work by utilizing JSONP to overcome these limitations. The resulting client-side JavaScript library has been published as DSLib [Inst00]. This library is used by the project *Open Source Registry System for Rare Diseases in the EU* (*OSSE)* for seamlessly integrating two data pools [MLWÜ14, MuLÜ14]. Moreover, the system uses the identity management component by Lablans et al. [LaBÜ15]. The system provides tight coupling of this component with a newly developed EDC system for clinical data. All of these solutions do not support two-tier pseudonymity as shown in [LKPK15, PKLK15]. Moreover, they focus on integrating master data with clinical phenotype data, whereas our solution is more generic. In our research networks we integrate various types of complex data in several different views (of course, only if permitted and required).

On an application level, pseudonymized research data management systems can be distinguished by whether they implement centralized or decentralized (de-) pseudonymization of data and whether information from different partitions is integrated on data layer, presentation layer or functional layer. These design alternatives can also be assessed from a privacy perspective. Furthermore, different designs are differently well suited for implementing the techniques described in this work. For example, some approaches utilize pseudonymization to integrate off-the-shelf applications into research system, e.g. [PrRi11]. A potential benefit of this approach is that the technology underlying the individual systems is often very heterogeneous which makes it difficult for an attacker to breach multiple components. In contrast, this directly leads to a functional distribution of data. It is therefore very important that the individual systems employ effective mechanisms to protect their data from various privacy risks. In contrast, a research system that is implemented from scratch allows incorporating a consistent system-wide privacy-preserving partitioning scheme from the beginning (*privacy by design*).

Multiple privacy concepts employing pseudonymization have been discovered in literature. De Moore et al. proposed a pseudonymization scheme utilizing a TTP to maintain mappings between identifying and medical data at data sources and data registries [MoCM03]. Kalra et al. focused on generic methods for developing privacy-preserving integrated biomedical data repositories for scientific research, which were exemplified with a cancer care scenario [KSMM05]. Pommerening et al. proposed a generic data protection scheme [PRDS05]. Neubauer et al. presented a generic and patient-controlled pseudonymization concept for primary and secondary use of health care data [NeHe11]. The ISO 25237 technical specification, which has been summarized in [MeMR08], aims at providing a comprehensive conceptual and practical guideline for implementing pseudonymization technologies [Inte00].

None of these proposals builds upon a comprehensive model. Most of them do not discuss even basic threats [KSMM05, MoCM03, PRDS05]. A threat analysis is outlined in [NeHe11], but the scenarios are not focused on privacy. The work from [Inte00] presents an isolated threat model as a starting point for risk analyses but it is not described how this model is related to the concept and how it can be used to build secure systems. “Most articles only consider high-level models of security and privacy that intermingle aspects of protecting data at rest, data flow and authorization” [PKLK15] (e.g., gradual access to microdata, aggregate data and anonymous data) [KSMM05, MoCM03, NeHe11, PRDS05].

3.3.3 Limitations[54]

Access statistics[55] for our applications show that about 25% percent of our users still access the systems with legacy browsers, such as Internet Explorer 8. As a consequence, we decided to implement our solution with technologies that are supported in older versions of widespread web browsers and did not utilize modern HTML 5 features, such as CORS, or client-side frameworks for building Single-Page Applications, such as AngularJS. Compared to the technologies currently utilized in our solution, these methods have a great potential to reduce system complexity. The main reason is that instead of distributing business logic over several backend servers, more functionality can be bundled into the client application, reducing the need for logic that orchestrates distributed operations. Moreover, application development and system maintenance are simplified, because the complexity of the backend services can be reduced to a minimum. As support for modern HTML features

[54] Cf. “Limitations” in [LKPK15].

[55] i.e. from July 2015

increases, we plan to upgrade our solution from a tightly coupled application with server-side rendering to a single-page application.

The general problem is that it remains unclear, how exactly data is to be separated into subsets. The ad-hoc classification into "identifying data" and "other types of data" is insufficient. For example, it is well understood that data which may fall into the second category can be used to re-identify individuals (see [LoDM10] for a discussion regarding diagnosis codes). To the best of our knowledge, ISO 25237 [Inte00] is the only work in the context of pseudonymization that lists a set of common identifiers with a high risk of re-identification. But still, no countermeasures against this inherent problem of pseudonymity have been proposed. This situation makes it difficult to find an adequate balance between privacy concerns and support for workflows that require re-identification of data and subjects. For example, the pseudonymization and de-pseudonymization process may be designed differently. The work by Aamot et al. [AKRK13] suggests an efficient routine process that requires to contact multiple selected persons called "ombudsmen", each of which controls a horizontal subset (i.e. data about a certain set of patients) of the data, to de-pseudonymize datasets. In contrast, the concept of Pommerening et al. [PDHG14] involves two additional parties in the process of de-pseudonymizing research data, each of which controls a vertical subset of the data (i.e. a certain set of attributes for all patients).

3.3.4 Conclusions[56]

A fully web-based single-page application design as illustrated in Figure 16 can be realized as a client-side JavaScript Application that communicates cross-domain with REST-based services exchanging JSON-formatted data with the client. Since todays JavaScript frameworks have overcome the drawbacks of the past, this technology is well suited for the purpose of disease networks.

[56] Cf. "Conclusions" in [LKPK15].

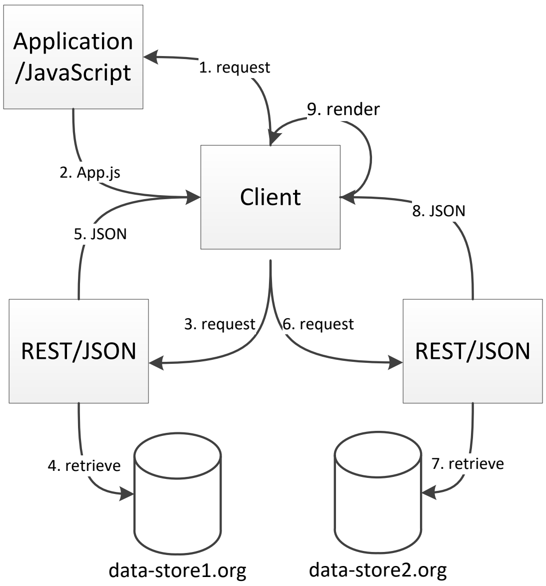


Figure 16. Single-page application design[57]

The frameworks have improved in terms of maturity, support and they have been employed for multiple every day applications. Advantageous for the implementation of a disease network with a JavaScript framework would be that server-side application business logic could be held simple. A lightweight JavaScript-Application would run on the client with the data seamlessly integrated a priori. Disadvantageous however, is the fragmented IDE support for JavaScript-frameworks which impedes development, depending on the technology employed. JavaScript is interpreted differently depending on the web browser. Therefore, server-side calculation and rendering could be considered more predictable.

Pseudonymization models are very heterogeneous, already on a conceptual level. Most importantly it remains unclear how exactly data is to be separated into distributed subsets. What is lacking is a thorough risk and threat analysis for pseudonymization schemes, covering at least the data- and the application level. Different architectural solutions exist for managing a set of pseudonymized data subsets, each of which has different properties in terms of usability, support for functional requirements and software complexity. Additionally, these architectures can be implemented with different technologies. In this work, we have analyzed this broad spectrum of architectural options and implementation

[57] Cf. [PKLK15]

techniques and we have presented a solution that is generic because it is independent of the actual distribution of data and supports a large set of features. In the future, we will investigate how using more modern HTML features can help to reduce system complexity and thus simplify application development as well as system maintenance.

4 Risk and Threat Analysis for the Reference Architecture

In this chapter, we will examine our reference architecture presented in chapter 3 for a detailed risk and threat analysis.

4.1 Background

Biomedical data is often personal and highly sensitive. National laws, e.g., the *HIPAA Privacy Rule* [Savo96], and international regulations, e.g., the General Data Protection Regulation of the European Union [Jour16], mandate stringent protection of such data. As a result, countermeasures must not only be implemented to mitigate security threats but additional measures need to be employed to mitigate *privacy threats.* Often a combination of different techniques is used. Important security measures include using protected network communication and strong authentication mechanisms to prevent unauthorized access by attackers [PKLK15].

Classical *security threats* include unauthorized access to data or manipulation of data and disruption of service. In this context, common countermeasures include using protected network communication, strong authentication mechanisms, redundant hardware, secure software development processes and clearly defined operating procedures. Systematic methodologies exist that help to consider security threats, risks and countermeasures throughout the whole software lifecycle, e.g., ISO 27001 [Iso16], Microsoft's *STRIDE* [HoLi06, MHLO14] and *DREAD* [Msdn11]. These aim at protecting systems from classical threats, such as unauthorized access to or manipulation of data and ensuring the availability of the system. In this context, common countermeasures include using protected communication between components of the IT system and the user, fine-grained access control, recording the users' actions, redundant hardware, secure software development processes and clearly defined operating procedures [PKLK15].

4.1.1 Breaches of Privacy and Confidentiality

While confidentiality is related to data and can be seen as “an agreement about maintenance and who has access to identifiable data” [Schu99], privacy is related to persons as “a sense of being in control of access that others have to ourselves” [Schu99].

According to [Shir07], a core element of privacy is “*[...] the degree to which [one] is willing to share [...] personal information*”. The biggest threat to privacy is access to personal data without the consent of the data owner. Authorization should directly reflect a person’s degree of consent for sharing data. In order to ensure privacy, confidentiality needs to be implemented. Privacy-enhancing techniques (PET) are utilized to establish multiple levels of defense that an attacker needs to overcome in order to get access to personal data. Pseudonymity can be considered an additional measure to classical security measures: Data is distributed to multiple pools. Ideally, each pool contains only personal data of limited value to an attacker. If an attacked successfully gains access to one pool, it will not necessarily lead to disclosure of an identity or essential information on such. The challenge is to separate the data accordingly and to determine which attributes will be combined in one pool. This separation of data is still not driven by proven concepts from research [PKLK15]. Personal information[58] and the protection of it “is the aspect that a direct relation between some data and a person must not be revealed. This also means that a person must not be identifiable by characteristics of the given data” [PKLK15] (i.e. data contained in one pool).

In [EmAr13] four plausible attacks on a data set were sketched:

1. Deliberately re-identification of data by recipient.
2. Spontaneous re- identification of data by recipient.
3. Data breach at recipient’s site.

[58] “[…], especially information […] that could cause harm or pain to that person if disclosed to unauthorized parties […]” [Shir07]. Examples include: “Any information a) that identifies or can be used to identify, contact, or locate the person to whom such information pertains; b) from which identification or contact information of an individual person can be derived; or c) that is or can be linked to a natural person directly or indirectly “[Itut10].

4. Demonstration attack on data launched by an adversary.

Liu et al. have reported over 900 data breaches from 2010 to 2013 affecting 29 millions of records only in the United States [LiMC15]. Application-layer pseudonymity in biomedical research aims at preventing breaches of privacy resulting from one or multiple attacks. We define an attack[59] as an unauthorized access to data, e.g., by compromising a system. Every successful attack results in a breach of confidentiality but not necessarily in a breach of privacy. *Privacy enhancing technologies* (PET) do not primarily aim at avoiding attacks but try to restrict privacy breaches. A privacy breach means that the data obtained by a breach of confidentiality allows one or multiple individuals to obtain new knowledge about one or multiple identified individuals. This can be a direct or indirect process, involving several actors of which not each needs to be an attacker. Actors collected data with their own knowledge and pass it on to others until the dataset enables someone to recognize one or more contained individuals.

4.1.2 Security Measures

These efforts are often supported by IT systems, which manage highly sensitive personal health information. Increasing public awareness of privacy threats has led to social and political pressure to prevent the misuse of personal data. As a result, national laws and international regulations mandate stringent data protection. Examples include the *HIPAA Privacy Rule* [Savo96] or the General Data Protection Regulation of the European Union [Jour16]. Both require IT systems to fulfill all state-of-the-art security properties, as, e.g., specified by ISO/IEC 27000 [Isoi09]. These include guaranteeing *confidentiality* and *integrity*, e.g., by exclusively using protected communication. Strong *authentication* methods and fine-grained *authorization* mechanisms must be implemented, e.g., using a role-based model. *Accountability* must be ensured, e.g., by logging the users' actions in an audit trail. Redundant hardware and intrusion detection systems help to ensure *availability*. A secure software development process with extensive testing reduces

[59] Any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself [Syst10].

software vulnerabilities[60]. On an organizational level, security threats can be mitigated by educating users and developing clearly defined operating procedures [PKLK15].

4.1.3 Privacy Measures

Privacy measures include de-identification and pseudonymization. *De-identification* methods remove or alter potential identifiers. The aim is to permanently and irreversibly *"remov[ing] the association between the identifying data set and the data subject."* [Inte00] making re-identification difficult for attackers. In contrast, *pseudonymity* is a reversible measure that inserts separative measures against attackers [PKLK15].

To protect datasets from privacy threats, different techniques are most relevant in different phases of our process from Figure 1. *De-identification* is the most important privacy measure when releasing and sharing research data. Examples include de-identification according to the HIPAA privacy rule [Savo96], restriction to HIPAA's limited data set, and statistical methods, such as k-anonymity [Swee02]. These methods introduce fuzziness, e.g., via generalization or suppression, and therefore allow balancing the level of detail and identifiability. The metadata that is used to provide other researchers with an overview of existing data in the second phase is often created by means of *aggregation*. Again, this allows balancing the level of detail and identifiability as individual-level data items are summarized and counts are provided that do not allow identifying specific individuals.

Pseudonymization as a Privacy Measure

Apart from standard security measures, separate storage of different types of data is an increasingly popular method to mitigate privacy risks [PKLK15]. For example, the *European General Data Protection Regulation* requires that *"[…] data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately […]"*[Jour16].. This is commonly known as *pseudonymization*, which aims at preventing attackers from relating research data to individuals. These sensitive relationships are protected with *pseudonyms*, which are

[60] "Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source" [Syst10].

artificial identifiers for which the link to the corresponding entity is kept *confidential* [Itut10].

4.2 Methods

In this chapter we will perform a threat analysis according to well-known methodologies (STRIDE, LINDDUN). The STRIDE [MHLO14] methodology designed to model security threats. This is complemented with the LINDDUN [DWSP11] methodology that focuses on privacy threats. This process consists of several steps that range from the definition of use scenarios, to the modelling of threats and risks to appropriate countermeasures. We will explain the details in the following.

Threat Modelling using STRIDE

The STRIDE [MHLO14] methodology provides an appropriate means to analyze threats and countermeasures systematically. STRIDE is an acronym for the security threat types addressed by the methodology which are (1) *spoofing*, (2) *tampering*, (3) *repudiation*, (4) *information disclosure*, (5) *denial-of-service*, and (6) *elevation-of-privilege*. We will relate these principles to the basic security principles of ISO 27000 [Isoi09] and RFC-4949 [Shir07]:

1. “Authenticity – property that an entity is what it claims to be” [Isoi09]
2. ”Integrity – property of protecting the accuracy and completeness of assets” [Isoi09]
3. “Accountability – responsibility of an entity for its actions and decisions” [Isoi09]
4. “Confidentiality – property that information is not made available or disclosed to unauthorized individuals, entities, or processes” [Isoi09]
5. “Availability – property of being accessible and usable upon demand by an authorized entity” [Isoi09]
6. “Authorization – approval that is granted to a system entity to access a system resource” [Shir07]

Privacy Threat Modelling using LINDDUN

LINDDUN is an extension to STRIDE suggested by Deng et al. [DWSP11] and allows for modeling privacy threats software systems. LINDDUN supports the identification of threats based on information flows and enables the selection of suitable countermeasures. LINDDUN is acronym that is formed by the basic privacy threat types: **L**inkability, **I**dentifiability, **N**on-repudiation, **D**etectability, **D**isclosure of information, Content

unawareness, and Policy and consent **n**on-compliance. These threat types are defined as follows:

Table 10. LINDDUN threat types

#	Type	Description
1	Linkability	"Linkability of two or more items of interest (IOI) (IOIs, e.g., subjects, messages, actions, etc.) allows an attacker to sufficiently distinguish whether these IOIs are related or not within the system"[DWSP11].
2	Identifiability	"Identifiability of a subject means that the attacker can sufficiently identify the subject associated to an IOI"[DWSP11].
3	Non-repudiation	"Non-repudiation allows an attacker to gather evidence to counter the claims of the repudiating party, and to prove that a user knows, has done or has said something"[DWSP11].
4	Detectability	"Detectability of an IOI means that the attacker can sufficiently distinguish whether such an item exists or not"[DWSP11].
5	Information disclosure	"Information disclosure threats expose personal information to individuals who are not supposed to have access to it"[DWSP11].
6	Content unawareness	"Content unawareness indicates that a user is unaware of the information disclosed to the system"[DWSP11].
7	Policy and consent non-compliance	"Policy and consent non-compliance means that even though the system shows its privacy policies to its users, there is no guarantee that the system actually complies to the advertised policies"[DWSP11].

The privacy properties these threats compromise are, respectively:

Table 11. Privacy properties

#	Type	Description
1	Unlinkability	"Unlinkability of two or more IOIs … means that within the system …, the attacker cannot sufficiently distinguish whether these IOIs are related or not" [PfHa10].

2	Anonymity	“Anonymity of a subject … means that the attacker cannot sufficiently identify the subject within a set of subjects, the anonymity set” [PfHa10].
3	Pseudonymity	“Pseudonymity – A subject is pseudonymous if a pseudonym is used as identifier instead of one of its real names” [PfHa10].
4	Plausible deniability	“Plausible deniability … means that an attacker cannot prove a user knows, has done or has said something” [DWSP11].
5	Undetectability and unobservability	“Undetectability and unobservability … of an IOI … means that the attacker cannot sufficiently distinguish whether it exists or not” [PfHa10].
6	Confidentiality	“Confidentiality means preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information” [McGK10].
7	Content awareness	“Content awareness – The user needs to be aware of the consequences of sharing information” [DWSP11].
8	Policy and consent compliance	“Policy and consent compliance ensures that the system’s (privacy) policy and the user’s consent … are indeed implemented and enforced” [DWSP11].

4.3 Results

4.3.1 Threat Modelling using LINDDUN/STRIDE

According to Deng et al. STRIDE modeling process can be substructured into the following nine steps [DWSP11]:

1. Definition of use scenarios where the key functionality is described.
2. Definition of external dependencies of the system to examine.
3. Definition of security assumptions of the system.
4. Identification of external security dependencies and restrictions.
5. Creation of data flow diagrams in order to analyze the system.
6. Determination of threat types according to STRIDE taxonomy.
7. Identification of threats using the data flow diagram.
8. Determination of risk levels for threats identified.
9. Planning risk mitigation by introducing countermeasures.

We will follow these steps in order to perform our analysis of threats and countermeasures.

4.3.1.1 Definition of Use Scenarios

We described a generic solution for implementing pseudonymized data management for a common and typical scenario: collaborative electronic collection of biomedical research data and further payload data, such as metadata about associated entities (e.g. biospecimen). Fort a detailed definition of use scenarios, we refer to section 1.1.2 and chapter 2.

4.3.1.2 External Dependencies

In this section we will define the external dependencies of our solution. External dependencies are the components on which our solution relies (i.e. technology stack). These components range from the operating system of a server to the database used. In Table 12 we listed the most important dependencies in our context.

Table 12. External dependencies

Type	Description
Operating system	Windows Server 2008 R2
Operating system	Ubuntu 12.x LTS
Webserver	Apache 2.x
Application Server	Tomcat 7.x
Java Version	Java 8.x
Database	MySQL (5.x)
Frameworks & Libraries[61]	e.g. Hibernate, Richfaces

4.3.1.3 Security Assumptions

In this section the basic security assumptions are defined. In the context of our system, these assumptions comprise the security requirements from chapter 2. These requirements are listed in the following tables (cf. Table 13, Table 14, Table 15).

Table 13. Physical security requirements

Req.ID	Requirement	Description
PSR-1	Physical access restrictions to client and server hardware	"Servers must be housed within a dedicated locked room with unescorted access limited to specified individuals" [OKCL11].
PSR-2	Secured power supply	"The power supply to servers should be secured, e.g. by a UPS unit, to allow an orderly shutdown on power failure" [OKCL11].
PSR-3	Encryption of non-physically secure data	"No patient data should be stored on anything other than protected servers (e.g. on laptops, desktops, USB sticks etc.) unless it is encrypted" [OKCL11].
PSR-4	Server failure - response	Alerts on server failure should be sent automatically to relevant personnel [OKCL11].

[61] For the sake of brevity, we will not list all libraries and frameworks that were employed. An exhaustive analysis would need to take all components into consideration.

PSR-5	Controlled environment	“Servers should be housed in a temperature controlled environment” [OKCL11].
PSR-6	Server room/building linked to response centers	“The server room/building should have an alarm system with the alarm linked to a central response center” [OKCL11].
PSR-7	Hazard control - fire alarms	“The server room should be fitted with heat and smoke alarms, monitored 24/7” [OKCL11].
PSR-8	Hazard Control - fire response	“The server room should be fitted with automatic fire response measures (e.g. inert gas)” [OKCL11].
PSR-9	Physical separation of data	The system shall support the hosting of different backends on different physical machines with different host names [PDHG14].
PSR-10	Separation of powers and duties	Servers hosting the different backend systems must be spatially and organizationally separated. This comprises as well different rooms and staffing for the separated backend servers [PDHG14].

Table 14. Logical security requirements (I)

Req.ID	Requirement	Description
LSR-1	Security management system	Regular reviews of IT security systems, practices and documentation, […], should occur as part of an ongoing Security Management System” [OKCL11].
LSR-2	Commitment to data protection	A data protection officer shall keep watch over relevant security policies and trainings [OKCL11, PDHG14].
LSR-3	External firewalls	External firewalls should be in place and configured to block inappropriate access [OKCL11, PDHG14].
LSR-4	Encrypted transmission	All data transmitted over the internet must be encrypted with state of the art encryption technology, e.g., TLS with server certificates, SHA-256 encryption for files [OKCL11, PDHG14].
LSR-5	Server admin role	“Servers should be protected by a highly restricted administrator password (i.e. known to essential systems staff only)” [OKCL11].
LSR-6	Admin password management	“The administrator password should be changed regularly according to locally agreed policies, and stored securely for emergency use (e.g. off-site)” [OKCL11].
LSR-7	Server maintenance	“Necessary patches and updates should be identified and applied in a timely but safe manner to: the operating system, anti-malware systems, backup systems and major apps (e.g. Clinical DBMSs, Web servers, Remote Access systems, etc.)” [OKCL11].
LSR-8	Commitment to information security	“The unit or its parent organization can demonstrate management commitment to information security, including relevant groups, policies, training and individuals with designated roles (e.g. 'IT security officer')” [OKCL11, PDHG14]

LSR-9	Internal firewalls	“Internal firewalls should be in place and correctly configured, e.g. blocking access to other departments, students” [OKCL11].
LSR-10	Security testing	“Regular security testing should be carried out and is documented” [OKCL11].
LSR-11	Intrusion detection with traffic monitoring	“Traffic activity should be monitored and hacking attempts identified and investigated” [OKCL11].
LSR-12	Logical access procedures	“SOPs and policies for access control to the network(s) and specific systems should be in place” [OKCL11].
LSR-13	Access control management	“Each system requiring access controls should have mechanisms, e.g. using roles, group membership, etc., that can be used to effectively differentiate and manage access” [OKCL11, PDHG14].
LSR-14	Granularity of access	“Access control mechanisms should be granular enough so that users only see the data they need to see” [OKCL11, PDHG14].

Table 15. Logical security requirements (II)

Req.ID	Requirement	Description
LSR-15	Password management	“Network password management should be enforced on all users, including regular password change and password complexity” [OKCL11].
LSR-16	Desktop lockout	“Desktop logins should post a blank screen or screensaver after a locally determined shut down period, and require password re-activation” [OKCL11].
LSR-17	Review of data and system access rights	“Access rights […] should be regularly reviewed, changes to access requested and actioned according to defined procedures, by designated individuals, with records kept of all rights, when granted, why and by whom.” [OKCL11].
LSR-18	Data security	All authorized personnel involved will keep data secure and confidential at all times [OKCL11, PDHG14].
LSR-19	System security	“System security and access control is ensured, data is only accessible to authorized personnel” [OKCL11].
LSR-20	Client-side re-combination	Systems need to be designed in a way that the reconstruction of the logical global dataset can *only be performed at the client-side* to reduce the number of attack vectors [PDHG14].
LSR-21	Confidentiality of internal identifiers	Clients must *not be able to learn the pseudonymous identifiers* used in the distributed databases. Pseudonymized datasets may only be joined by authorized users [PDHG14].
LSR-22	Two-tier pseudonymization	The system shall provide support for two-tier pseudonymization, implemented with an additional mapping service [Inte00, PDHG14].

LSR-23	Restriction of data access	The system shall support a site-based view where a physician and staff have access restricted to data of their site. The system shall support researchers having access restricted to data concerning biospecimen. Furthermore, the system shall support data managers having access to patients medical data without having access to identification data [OKCL11, PDHG14].
LSR-24	Logging procedures	Every request to a system must be logged in a separate file [OKCL11, PDHG14].

4.3.1.4 External Security Notes

As already noted, our solution utilizes several external components (cf. Table 12). Regarding security implications of external dependencies, we oriented towards the Security Technical Implementation Guides (STIGs) listed in [Unif15]. STIGs list known vulnerabilities with weighted severity and therefore allow for minimization of vulnerabilities when using standard components. In the following, we will present a brief overview of the requirements concerning the external components utilized by our solution (cf. Table 16 and Table 17). We will only present the requirements weighted with a high severity. For a comprehensive list, we refer to [Unif15].

Table 16. External security requirements (I)

Req.ID	Ext. Component	Requirement
ESR-1	Apache 2.x	“Server side includes (SSIs) must run with execution capability disabled” [Unif15].
ESR-2	Apache 2.x	”All web server documentation, sample code, example applications, and tutorials must be removed from a production web server” [Unif15].
ESR-3	Apache 2.x	”Web server software must be a vendor-supported version” [Unif15].
ESR-4	Apache 2.x	”Administrators must be the only users allowed access to the directory tree, the shell, or other operating system functions and utilities” [Unif15].
ESR-5	Java 8.x	”Java Runtime Environment (JRE) versions that are no longer supported by the vendor for security updates must not be installed on a system” [Unif15].
ESR-6	Windows Server 2008 R2	”Systems must be at supported service pack (SP) or release levels” [Unif15].
ESR-7	Windows Server 2008 R2	”Named pipes that can be accessed anonymously will be configured to contain no values” [Unif15].
ESR-8	Windows Server 2008 R2	”Anonymous enumeration of SAM accounts will not be allowed” [Unif15].

ESR-9	Windows Server 2008 R2	”Anonymous enumeration of shares will be restricted” [Unif15].
ESR-10	Windows Server 2008 R2	”Autoplay will be disabled for all drives” [Unif15].
ESR-11	Windows Server 2008 R2	”The LanMan authentication level will be set to Send NTLMv2 response only\refuse LM & NTLM” [Unif15].
ESR-12	Windows Server 2008 R2	”The Recovery Console option will be set to prevent automatic logon to the system” [Unif15].
ESR-13	Windows Server 2008 R2	”Anonymous access to Named Pipes and Shares will be restricted” [Unif15].
ESR-14	Windows Server 2008 R2	”Local volumes will be formatted using NTFS” [Unif15].

Table 17. External security requirements (II)

Req.ID	Ext. Component	Requirement
ESR-15	Windows Server 2008 R2	”Unauthorized accounts must not have the Debug programs user right” [Unif15].
ESR-16	Windows Server 2008 R2	”Unauthorized accounts must not have the Create a token object’ user right” [Unif15].
ESR-17	Windows Server 2008 R2	”The Enhanced Mitigation Experience Toolkit (EMET) v5.x or later must be installed on the system” [Unif15].
ESR-18	Windows Server 2008 R2	”The Windows Installer Always install with elevated privileges must be disabled” [Unif15].
ESR-19	Windows Server 2008 R2	”Unauthorized remotely accessible registry paths must not be configured” [Unif15].
ESR-20	Windows Server 2008 R2	”The default autorun behavior will be configured to prevent autorun commands” [Unif15].
ESR-21	Windows Server 2008 R2	”Solicited Remote Assistance will not be allowed” [Unif15].
ESR-22	Windows Server 2008 R2	“The system will be configured to prevent the storage of the LAN Manager hash of passwords” [Unif15].
ESR-23	Windows Server 2008 R2	”Network shares that can be accessed anonymously will not be allowed” [Unif15].

ESR-24	Windows Server 2008 R2	”The use of local accounts with blank passwords will be restricted to console logons only” [Unif15].
ESR-25	Windows Server 2008 R2	”Unauthorized remotely accessible registry paths and sub-paths must not be configured” [Unif15].
ESR-26	Windows Server 2008 R2	”Unauthorized accounts must not have the Act as part of the operating system user right” [Unif15].

4.3.1.5 Data Flow Diagram

Data flow diagrams can be used to show the relationships between and among processes and data. We will start with describing the elements used in the data flow diagram.

Table 18. Components of the data flow diagram

Graphical Notation	Name (Acronym)	Description
DS	Data store (DS)	Repositories of data in the system [Vie00].
→	Data flow (DF)	Data flow of the physician’s (or staff) workflow.
→	Data Flow (DF)	Data flow of lab personnel.
•••••▶	Material flow (MF)	Flow of physical material (e.g. biospecimen, paper documents).
P	Process (P)	“A process transforms incoming data flow into outgoing data flow” [Vie00].
E	Entity (E)	“External entities are sources and destinations of the system's inputs and outputs” [Vie00]. In our context, an entity is a user of the system.
EE	External entity (EE)	In our context, external entities are patients about whom data is collected.
⛁	Permanent storage (PS)	This element illustrates a permanent storage of information.
⟩	Temporary storage (TS)	Data is only available for a limited time (e.g. user session).
⚷	Pseudonymization (PZ)	Translation of one identifier into another.
🗋	Paper document (PD)	Paper documents important in the process (e.g. informed consent).

⬚	Trust boundary (TB)	Trusted hard- and software infrastructure.

In Table 18 the elements we used for modeling the data flow diagram can be seen. We introduced further custom elements for a more detailed illustration of the workflow. The result is shown in Figure 17.

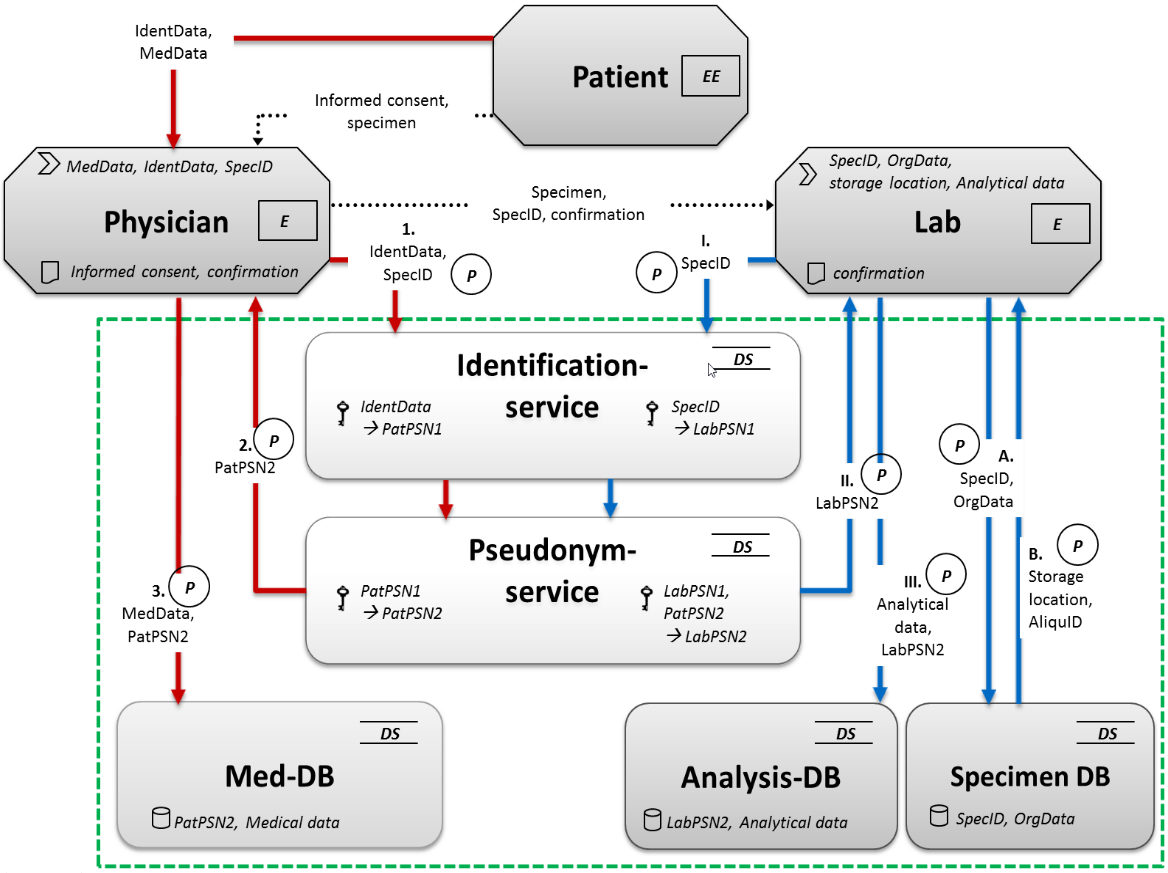


Figure 17. Data flow in the system[62]

[62] A similar diagram has been published in [Fkoh10]

4.3.1.6 Identification of Threats

In Table 19 we show the mapping of basic security properties from ISO 27000 [Isoi09] and RFC-4949 [Shir07] to STRIDE security threats. Furthermore, we show the data flow elements from Table 18 that are related to STRIDE security threats as well. An "X" marks a relation of a data flow element to a threat.

Table 19. Mapping STRIDE security threats to data flow elements[63]

Security property	STRIDE security threats	(E)	(EE)	(DS)	(P)	(PD)
Authentication	Spoofing	X	X	X	X	X
Integrity	Tampering		X	X	X	
Non-repudiation	Repudiation	X		X	X	X
Confidentiality	Information disclosure	X	X	X	X	X
Availability	Denial of service			X	X	
Authorization	Elevation of Privilege	X			X	

The relation of data flow elements listed in Table 18 to LINDDUN privacy threats are shown in Table 20. The symbol „*X*“ indicates a potential privacy threat for a data flow element in the system.

Table 20. Mapping LINDDUN privacy threats to data flow elements

LINDDUN privacy threats	**Examples** [DWSP11]	(E)	(EE)	(DS)	(P)	(DF)	(PD)
Linkability	An attacker can relate different data items to an item of interest.	X	X	X	X	X	X
Identifiability	An attacker can identify a single data entity in a set of entities.	X	X	X	X	X	X
Non-repudiation	An attacker can undeniably verify that certain actions have been performed by an entity.			X	X	X	X
Detectability	Possibility for an attacker to distinguish whether data related to a particular entity exists or not.			X	X	X	X

[63] Cf. Data flow elements from Table 18

Information disclosure	Exposure of information to unauthorized entities.	X	X	X	X	X	X
Content unawareness	The data subject is unaware of the data processed, stored or deleted.	X	X				
Policy/consent noncompliance	The system or single components are not compliant with policies or consent.			X	X	X	X

In compliance with Deng et al. [DWSP11], we will now display the identified threats as threat trees. Each intersection marked with "X" of a data flow element and each threat can be represented by a dedicated threat tree. We modeled the corresponding trees for each intersection in the following.

Table 20. Data flow elements for threat trees[64]

Graphical Notation	Description
⬭	Root threat
▢	Concrete threat
→	Relation

[64] Cf. Figure 2 in [DWSP11]

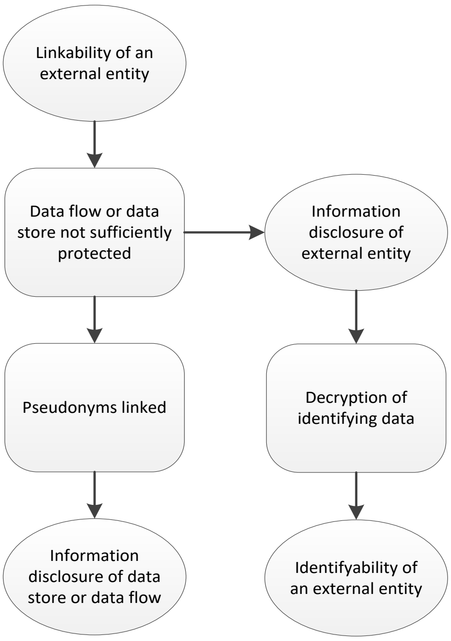

Linkability of an external entity
Data flow or data store not sufficiently protected
Information disclosure of external entity
Pseudonyms linked
Decryption of identifying data
Information disclosure of data store or data flow
Identifyability of an external entity

Figure 18. Threat tree for linkability and identifiability of an external entity

Linkability of an external entity (EE): „refers to an attacker can sufficiently distinguish whether two or more entities are related or not within the system [DWSP11].” The goal of an attacker is to disclose information of a data flow or a data store. In our context, this means that an attacker can learn about pseudonymous relationships of data items. The threat tree is shown in Figure 18. The preconditions are that data flow or data store are not fully protected or an attacker has successfully intercepted client-server communication. The information disclosure of an external entity can as well lead to identifiability of an external entity. Preconditions are the successful attack on a data store and the decryption of identifying data.

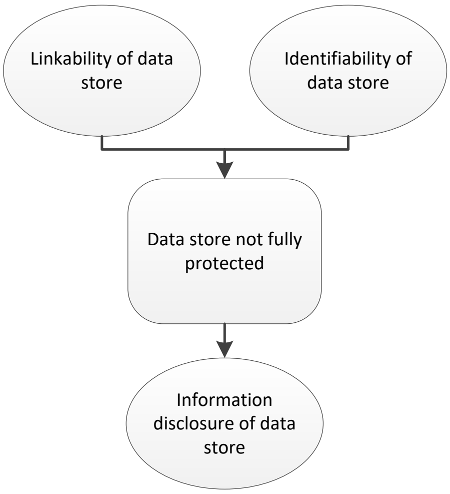

Linkability of data store
Identifiability of data store
Data store not fully protected
Information disclosure of data store

Figure 19. Threat tree for linkability and identifiability of data store

The goal of linkability and identifiability of a data store is to disclose information of a data store. Here, an attacker needs to exploit the vulnerabilities in the data store protection. A successful attack will lead to information disclosure of the data store. The threat tree is shown in Figure 19.

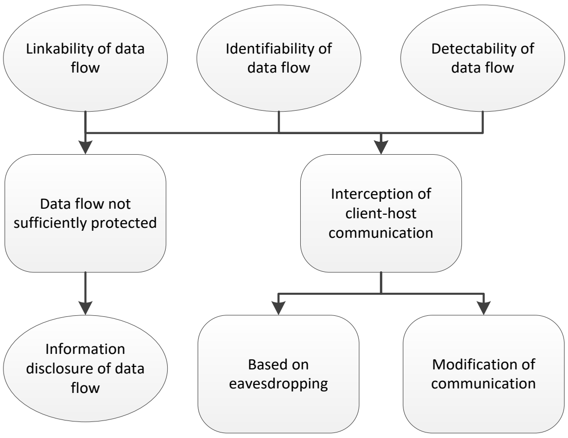

Linkability of data flow
Identifiability of data flow
Detectability of data flow
Data flow not sufficiently protected
Interception of client-host communication
Information disclosure of data flow
Based on eavesdropping
Modification of communication

Figure 20. Threat tree for a data flow

An attack on a data flow aims at disclosure of information of a data flow (cf. Figure 20). The precondition is that an attacker has intercepted client-host communication based on eavesdropping or modification of communication.

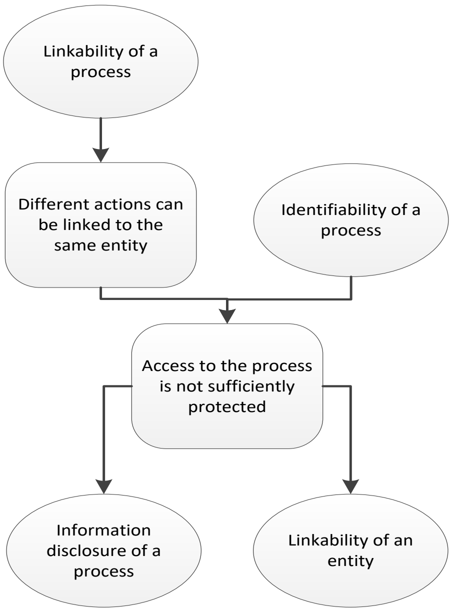

Linkability of a process
Different actions can be linked to the same entity
Identifiability of a process
Access to the process is not sufficiently protected
Information disclosure of a process
Linkability of an entity

Figure 21. Threat tree for a process

Linkability of a process (P): The threat tree shown in Figure 21 depicts the linkability of a process. The goal is to link multiple actions to an entity (i.e. user) and hereby disclose information about a process.

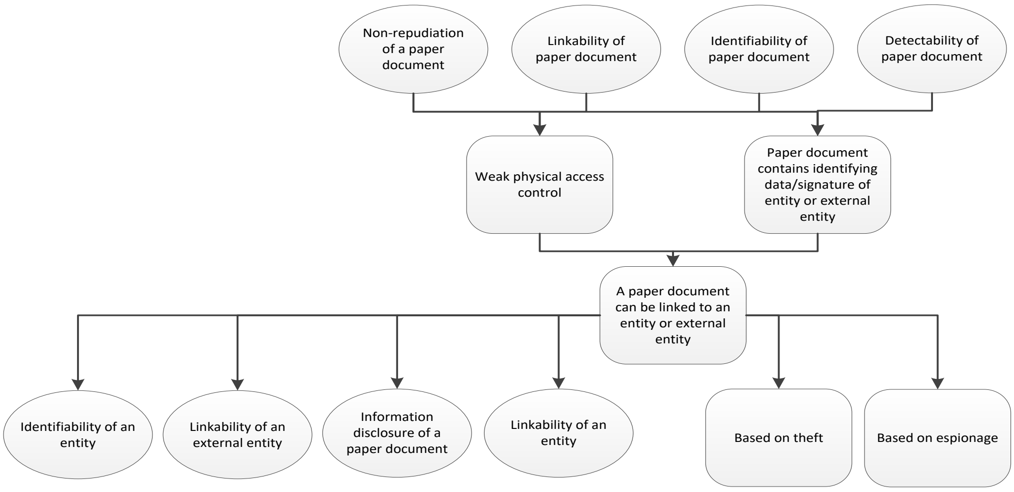

Non-repudiation of a paper document
Linkability of paper document
Identifiability of paper document
Detectability of paper document
Weak physical access control
Paper document contains identifying data/signature of entity or external entity
A paper document can be linked to an entity or external entity
Identifiability of an entity
Linkability of an external entity
Information disclosure of a paper document
Linkability of an entity
Based on theft
Based on espionage

Figure 22. Threat tree for a paper document

Linkability, detectability and non-repudiation of a paper document aim at information disclosure, linkability and identification of entities and external entities. The attack may be directed towards paper transfer routes or archives where paper based documents are stored by, e.g., theft or espionage. The threat tree is shown in Figure 22. Paper documents are likely to contain identifying information (e.g. informed consent) and therefore bear the risk of identifiability and linkability for entities or external entities. The attack may also lead to information disclosure of the paper document.

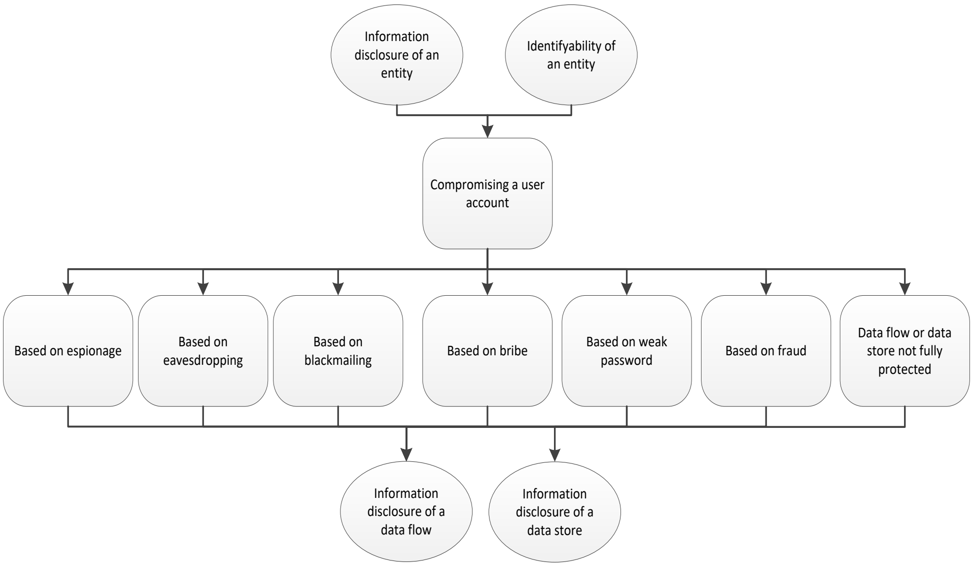

Information disclosure of an entity
Identifyability of an entity
Compromising a user account
Based on espionage
Based on eavesdropping
Based on blackmailing
Based on bribe
Based on weak password
Based on fraud
Data flow or data store not fully protected
Information disclosure of a data flow
Information disclosure of a data store

Figure 23. Threat tree for information disclosure and identifiability of an entity

The threat tree shown in Figure 19 depicts the preconditions of information disclosure and identifiability for an entity. In our context, entities are users of the system having access to certain information in data stores, depending on role-based access. The precondition is compromising a user account. This can be achieved based on espionage, eavesdropping, blackmailing, bribe, brute-force of passwords and fraud. Compromising a user account will lead to information disclosure of a data flow or data store.

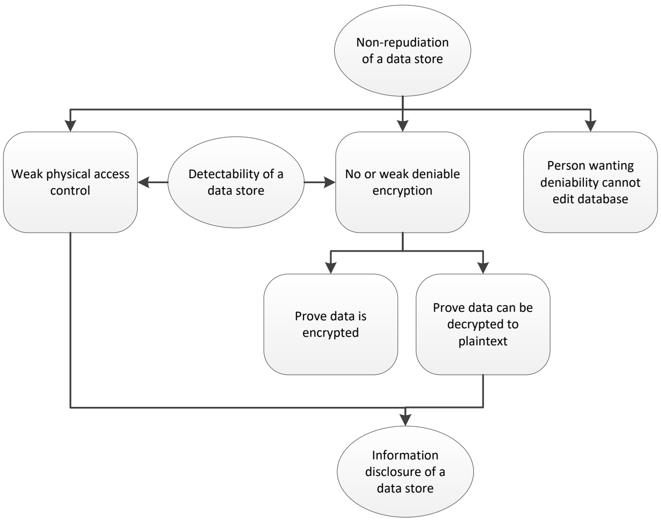

Non-repudiation of a data store
Weak physical access control
Detectability of a data store
No or weak deniable encryption
Person wanting deniability cannot edit database
Prove data is encrypted
Prove data can be decrypted to plaintext
Information disclosure of a data store

Figure 24. Non-repudiation and detectability of a data store

Non repudiation and detectability of a data store are closely related as shown in Figure 24. The preconditions for detectability are weak physical access control and weak encryption which can lead to information disclosure of a data store. An additional precondition for non-repudiation is that users of the system cannot edit the database in order to achieve deniability of their actions.

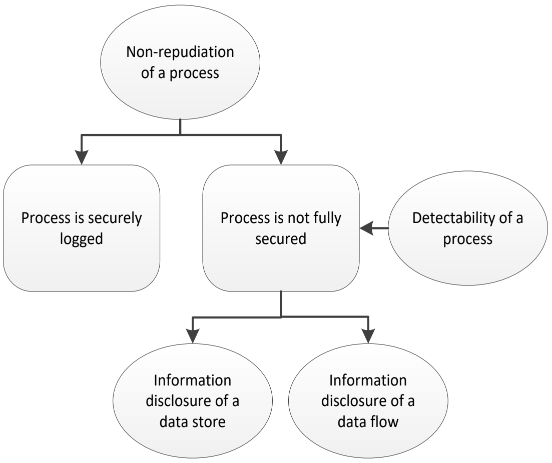

Non-repudiation of a process
Process is securely logged
Process is not fully secured
Detectability of a process
Information disclosure of a data store
Information disclosure of a data flow

Figure 25. Non-repudiation and detectability of a process

Non repudiation and detectability of a process share the precondition of a not fully protected process (cf. Figure 25). This can lead to information disclosure of a data flow or data store. Non-repudiation also results from a securely logged process.

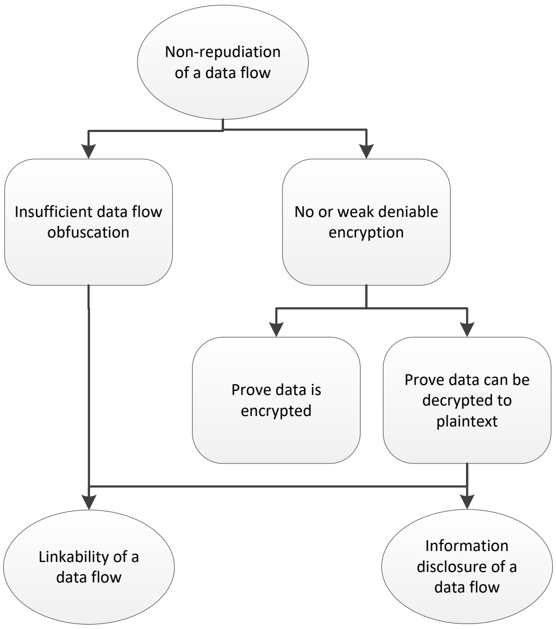

Non-repudiation of a data flow
Insufficient data flow obfuscation
No or weak deniable encryption
Prove data is encrypted
Prove data can be decrypted to plaintext
Linkability of a data flow
Information disclosure of a data flow

Figure 26. Non-repudiation of a data flow

Non-repudiation of a data flow has the preconditions of insufficient data flow obfuscation and no or weak encryption. Both can lead to linkability and information disclosure of a data flow.

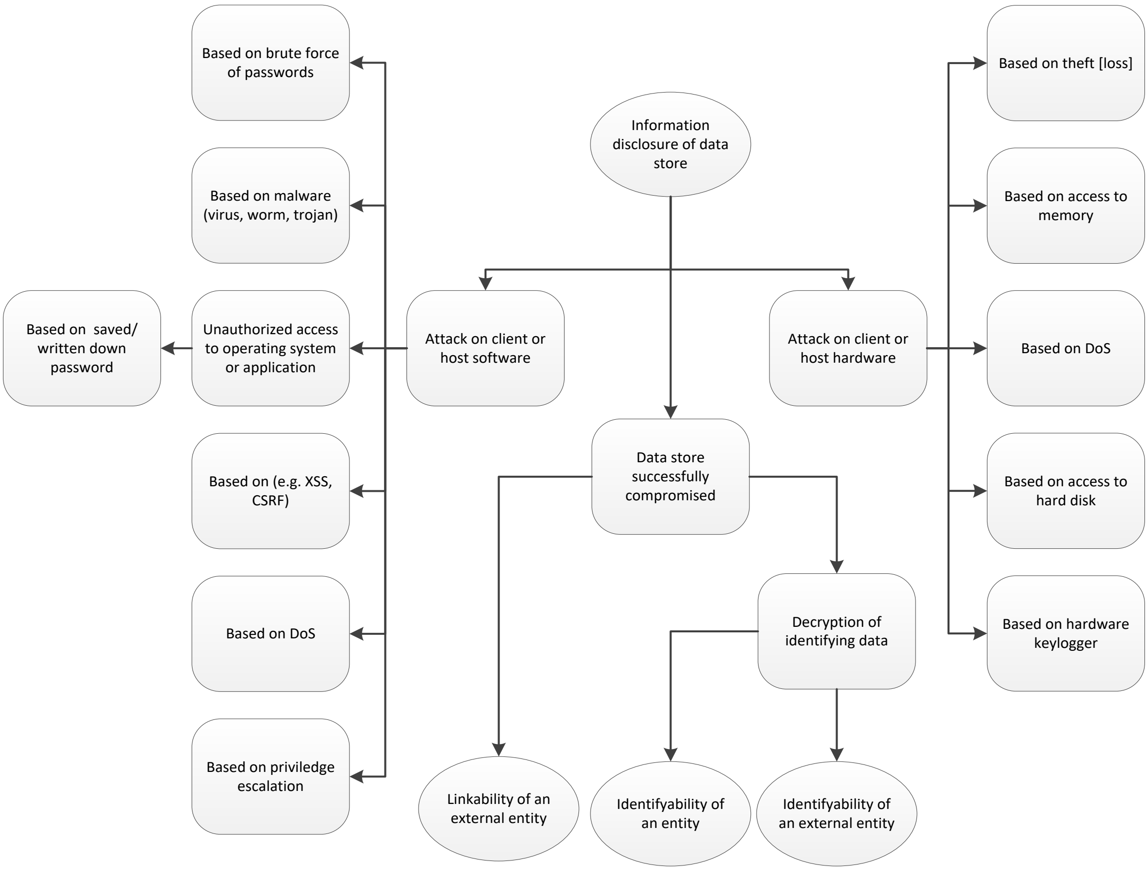

Information disclosure of data store
Attack on client or host software
Attack on client or host hardware
Data store successfully compromised
Based on brute force of passwords
Based on malware (virus, worm, trojan)
Unauthorized access to operating system or application
Based on saved/ written down password
Based on (e.g. XSS, CSRF)
Based on DoS
Based on priviledge escalation
Based on theft [loss]
Based on access to memory
Based on DoS
Based on access to hard disk
Based on hardware keylogger
Decryption of identifying data
Linkability of an external entity
Identifyability of an entity
Identifyability of an external entity

Figure 27. Threat tree for information disclosure of data store

Information disclosure of a data store (DS): This scenario requires the attacker to successfully compromise a data store. This can be performed either by attacking the data store hardware or data store software as depicted in Figure 27. This can lead to linkability or identifiability for an external entity as well as identifiability for an entity.

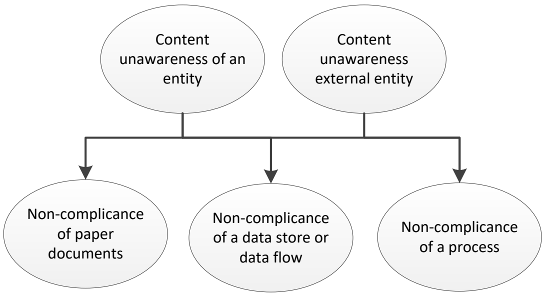

Content unawareness of an entity
Content unawareness external entity
Non-complicance of paper documents
Non-complicance of a data store or data flow
Non-complicance of a process

Figure 28. Content unawareness and policy/consent noncompliance

Content unawareness of entities and external entities and non-compliance of the other diagram components (paper documents, processes, data-stores, data flows) are closely related. Paper documents, (e.g. informed consent, data use policies) must be consistent with the workflows and processes inside a system. If this is not the case, workflows and processes may be non-compliant and/or entities may be unaware about the content of data stores, processes and data flow in the system.

4.3.1.7 Estimation of Risk

For the estimation of risk we followed the process described in NIST Special Publication 800-30 [Nist12]. This risk management process consists of four components:

1. Risk framing: Describes "the environment where risk-based decisions are made".
2. Risk assessment: Estimation of risk within the risk frame.
3. Risk response: Respond to risk based on risk assessment.
4. Risk monitoring: Observation of risk over time in the context of implemented risk responses.

According to [Nist12] the risk is a function of (i) the adverse impacts in case of occurrence, and (ii) the likelihood of occurrence. Next, we will describe the basic elements NIST utilizes for risk assessment.

- **Threat sources**: The intent, method or situation to exploit a vulnerability (e.g. a special type of attack).
- **Threat shifting:** Response of adversaries to countermeasures.
- **Vulnerabilities and predisposing conditions** that can be exploited by a threat source (i.e. technical and organizational).
- **Likelihood of occurrence**: "Probability that a given threat is capable of exploiting a given vulnerability (or set of vulnerabilities)" [Nist12]. For adversarial threats, the likelihood of occurrence is based on *intent, capability* and *target*. For other threat

events, the likelihood of occurrence is estimated using historical evidence, empirical data, or other factors [Nist12].

- **Impact**: “The level of *impact* from a threat event is the magnitude of harm that can be expected to result from the consequences of unauthorized disclosure of information, unauthorized modification of information, unauthorized destruction of information, or loss of information or information system availability” [Nist12].
- **Risk:** *“*is a function of the likelihood of a threat event’s occurrence and potential adverse impact should the event occur” [Nist12].

Table 21. Example template for risk assessment

Threat Event	Threat Sources	Vulnerabilities	Likelihood of Threat	Level of Impact	Risk	Countermeasures
STRIDE LINDDUN	- Malicious insider - Denial-of-service - Administrative error - Hardware fault - Power failure - etc.	- System security procedures - Internal controls - Lack of risk management - Poor intra-agency communications - External services - Not risk aware processes - etc.	low, medium, high	low, medium, high	low, medium, high	Countermeasures implemented

In Table 21 examples for the basic elements of risk assessment according to [Nist12] can be seen. The “threat event” column contains basic threats from STRIDE and LINDDUN as described previously. “Threat sources” comprises the origin of the attack as shown by the examples. The vulnerabilities column shows the component that an attack tries to exploit. We leave aside aspects like uncertainty, aggregation and threat shifting which do not apply in our context. For the assessment of likelihood of threat, the level of impact and finally the overall risk, we decided to utilize a qualitative scale comprising the values: *low*, *medium* and *high*. The countermeasures column describes the countermeasures implemented by the system to mitigate the attack.

Table 22. Level of risk assessment

Level of Risk			
Likelihood of Threat	Level of Impact		
	low	medium	high
Low	low	low	med
Medium	low	med	high
high	low	med	high

As can be seen in Table 22, we chose color coding for level of risk assessment. The function of “likelihood of threat” multiplied with “level of impact” results in the following results for assessment: The color *green* indicates a *low risk level, yellow* respectively a *medium risk level* and red a *high risk level* [PKLK15].

4.3.1.8 Description of Countermeasures of the Reference Architecture

We have described the countermeasure implemented in our solution in chapter 3. We will now list these and describe their features in the following:

Table 23. List of countermeasures (I)

ID	Countermeasure
CM-1	Auditing and logging
CM-2	Automatic logout after inactivity
CM-3	Automatic updates of external components (OS, JVM etc.)
CM-4	Backups/disaster recovery plan
CM-5	Client-side recombination of distributed data
CM-6	Confidentiality of internal identifiers
CM-7	Consent Management
CM-8	Data Use Agreements

Auditing and logging procedures (*CM-1*) are a common functionality in web-based registries, for it assures accountability and integrity. Every request is logged and every data modification recorded in the database. The automatic logout after inactivity (*CM-2*) is a standard mechanism that needs no further explanation. The timeout however should be determined by the use case specific needs and workflows. Automatic update of external components (*CM-3*) serves for closing newly discovered vulnerabilities of external components as soon as possible. Backups and a disaster recovery plan (*CM-4*) are essential in case of an attack or simply a hardware failure. The client-side recombination of distributed data (*CM-5*) as well as the confidentiality of internal identifiers (*CM-6*) is an aspect we have discussed in the previous chapter. Data use agreements represent a contract between the data controllers and a researcher to prevent the misuse of data and to report conspicuities of any kind. Consent management (*CM-7*) and Data Use Agreements (*CM-8*) are best practices in medical research networks which we will not explain in detail here.

Table 24. List of countermeasures (II)

ID	Countermeasure
CM-9	Database encryption
CM-10	Distributed authorization
CM-11	Encrypted backups
CM-12	Encrypted tokens for communication between backends
CM-13	Encryption of non-physically secure data
CM-14	Ethics committee review
CM-15	Firewalls and virus scanners (external and internal)
CM-16	Granularity of access adjustable

Database encryption (*CM-9*) and encrypted backups (*CM-11*) should protect against inside attacks of people who have access on the database level but it also protects against theft of hardware. Distributed authorization (*CM-10*) and encrypted tokens for communication (*CM-12*) were extensively described in the previous chapter (cf. chapter 3). The encryption of non-physically secure data (*CM-13*) is a very important aspect, even though it is not an aspect of our solution itself. When data is handed out via USB-sticks or other media, it must be encrypted. Ethic committee reviews (*CM-14*) should protect against workflows or functionalities that are not in line with the informed consent or other policies of a research endeavor. Firewalls and virus scanners (*CM-15*) on both, backends and client-workstations are obligatory. The granularity of access must be adjustable following the need-to-know principle.

Table 25. List of countermeasures (III)

ID	Countermeasure
CM-17	Input validation/sanitization practices
CM-18	Intrusion detection system with traffic monitoring
CM-19	IP-based filtering of requests
CM-20	Limit for login attempts
CM-21	Non-delegated authentication
CM-22	One-time access tokens
CM-23	Penetration testing
CM-24	Physical access restrictions to client and server hardware

Input validation/sanitization practices (*CM-17*) were also discussed in the previous chapter (cf. 3). An intrusion detection system with traffic monitoring (*CM-18*) is recommended to increase the level of network security. Traffic monitoring for backends allows for monitoring the web activity of backends as well as bandwidth and internet usage. The IP-based filtering of requests (*CM-19*) prevents data stores from receiving unauthorized requests. Communication within the trusted server environment requests are only allowed from known IP ranges. A limit for login attempts (*CM-20*) assures that an account is locked after a certain number of unsuccessful login attempts. After each unsuccessful attempt the response time of the backend increases exponentially. This mitigates brute force attacks on passwords. Non-delegated authentication (*CM-21*) as well as one-time access tokes (*CM-22*) were described in the previous chapter 3. Penetration testing (*CM-23*) of external components and the internal software stack is an obligatory measure. Physical access restrictions to client and server hardware ensures that no unauthorized person can access servers or client workstations.

Table 26. List of countermeasures (IV)

ID	Countermeasure
CM-25	Redundant server hardware/raid

CM-26	Role-based access control
CM-27	Secure server rooms including UPS and fire extinguisher
CM-28	Separation of powers and duties
CM-29	Server admin role and application admin role
CM-30	Admin password management
CM-31	Server rooms linked to response center
CM-32	Server hardening (Configuration management)

The usage of redundant server hardware (*CM-25*) assures fast recovery and fail safety. In connection with secure server rooms including UPS and fire extinguisher (*CM-27*) as well as server rooms linked to response centers (*CM-31*) represent state-of-the-art infrastructure. Role-based access control (*CM-26*) and the separation of powers and duties (*CM-28*) was described in the previous chapter (cf. chapter 3). This is also important in the context of server admin- and application admin roles (*CM-29*). Admin password management (*CM-30*) should assure secure handover and alternation according to agreed policies. Server hardening (*CM-32*) aims at reduction of vulnerabilities by minimization of software on a server, by closing ports not in usage, by regular revision of user accounts and permissions etc.

Table 27. List of countermeasures (V)

ID	Countermeasure
CM-33	Data access restrictions (Site-based view)
CM-34	Software installation policies
CM-35	Physical separation and spatial distribution of data
CM-36	Standard operating procedures (Patient information)
CM-37	Standard operating procedures (Data release)
CM-38	Standard operating procedures (Paper documents)
CM-39	TLS with server certificates

Data access restrictions in terms of a site-based view (*CM-33*) are a supplement of RBAC for assuring confidentiality. Software installation policies (*CM-34*) on client workstation as well as on backends are an important measure for mitigation of malicious software being installed. Physical separation and spatial distribution of data (*CM-35*) on different backends is also interlinked with the separation of powers and duties. Furthermore, it mitigates the impact of theft of single hardware components. Standard operation procedures for patient information, data release and paper documents are best practices and obligatory in a research network. TLS with server certificates (*CM-39*) represent state-of-the-art communication security for assuring confidentiality, integrity and authenticity.

Table 28. List of countermeasures (VI)

ID	**Countermeasure**
CM-40	Two-factor authentication
CM-41	Two-tier pseudonymization
CM-42	User account management policies
CM-43	User security and compliance trainings
CM-44	Username/password policies
CM-45	Virtualization
CM-46	Vulnerabilities of external components fixed

We have explained two-factor authentication and (*CM-40*) two-tier pseudonymization (*CM-41*) in the previous chapter (cf. chapter 3) extensively. User account management policies (*CM-42*) describe who is authorized to order the creation of new user account and who is allowed to execute it. Furthermore, it is defined who has the authority to create roles and determine the lifetime for an account. Security and compliance trainings for users (*CM-43*) are primarily directed against social attacks (e.g. bribe, fraud, blackmailing etc.). Username and password policies (*CM-44*) define how usernames and passwords are chosen and which rules apply. It also defines how often passwords must be altered which length and characters they must contain. Virtualization (*CM-45*) of server environment describes the usage of virtual machines emulating physical servers. This allows for a flexible configuration and fast disaster recovery.

4.3.1.9 Risk Assessment and Mitigation

In this section we will perform the assessment of likelihood impact and overall risk. We will show how we mitigate threats with implemented countermeasures shown in Table 23 - Table 28. For the assessment of risk we do not go into detail concerning countermeasures dealing with the exploitation of external components (cf. Table 12). This would exceed the limits of this work. For the list of countermeasures dealing with these threats, we refer to Table 16 and Table 17. The following risk assessment is performed as previously described in the template for risk assessment in Table 21. The acronym “L o T” and “L o I” stand for “Level of threat” and respectively “Level of Impact” (cf. Table 29 - Table 34). The risk is assessed according to Table 22.

Table 29. Risk assessment (I)

Scenario	**Threat Event**	**Threat Sources**	**Vulnerabilities and Predisposing Conditions**	**L o T**	**L o I**	**Risk**	**Countermeasures (Elements of the Security Architecture)**
User (Researcher) poses as something or somebody else	Spoofing, Information Disclosure	Malicious insider	Weak authentication system, poorly secured workstations, insufficient protection of system	LOW	MED	LOW	(1) Role-based access control, (2) Granularity of access adjustable, (3) Distributed authorization, (4) IP-based filtering of requests, (5) Limit for login attempts, (6) Encrypted one-time access tokens, (7) Two-factor authentication, (8) Data access restrictions (Site-based view), (9) Automatic logout after inactivity
User (physician or staff) poses as something or somebody else	Spoofing, Information Disclosure	Malicious insider	Weak authentication system, poorly secured workstations, insufficient protection of system	LOW	MED	LOW	(1) Role-based access control, (2) Granularity of access adjustable, (3) Distributed authorization, (4) IP-based filtering of requests, (5) Limit for login attempts, (6) Encrypted one-time access tokens, (7) Two-factor authentication, (8) Data access restrictions (Site-based view), (9) Automatic logout after inactivity

Unauthorized escalation of privileges for an account	Spoofing, Policy and Consent non-compliance Information Disclosure	External attacker, Malicious insider	Insufficient access control or granularity of access not adjustable, No RBAC	LOW	HIGH	MED	(1) Role-based access control, (2) Granularity of access adjustable, (3) Separation of powers and duties, (4) Data access restrictions (Site-based view), (5) Penetration testing, (6) User account management policies, (7) Distributed authorization
Sql Injection, Input validation/ sanitization failure, Over capacity failure	Tampering, Denial of Service	External attacker, Malicious insider	Missing Input validation/ sanitization, No server hardening	LOW	MED	LOW	(1) Input validation/sanitization practices, (2) Server hardening, (3) Penetration testing

Table 30. Risk assessment (II)

Scenario	**Threat Event**	**Threat Sources**	**Vulnerabilities and Predisposing Conditions**	**LoT**	**LoI**	**Risk**	**Countermeasures (Elements of the Security Architecture)**
Data flow between user and data stores (Man-In-The-Middle, Replay attacks)	Tampering, Detectability, Information Disclosure, Identifiability	External attacker	Insufficient secure connection	LOW	LOW	LOW	(1) TLS with server certificates, (2) IP-based filtering of requests, (3) Firewalls and virus scanners, (4) Non-delegated authentication, (5) Encrypted one-time access tokens
Data flow between lab workstation and data stores (Man-In-The-Middle, Replay attacks)	Tampering, Detectability, Information Disclosure, Identifiability	External attacker	Insecure data transfer	LOW	LOW	LOW	(1) TLS with server certificates, (2) Software installation policies, (3) Encrypted one-time access tokens, (4) Firewalls and virus scanners, (5) Non-delegated authentication
Manipulation of client software	Linkability, Information Disclosure, Identifiability	External attacker	No or weak workstation protection	MED	HIGH	HIGH	(1) Software installation policies, (2) Physical access restrictions, (3) Firewalls and virus scanners, (4) Role-based access control

Manipulation of host software	Linkability, Information Disclosure, Identifiability	External attacker, Malicious insider	No or weak protection of data stores	LOW	LOW	LOW	(1) Role-based access control, (2) Distributed authorization, (3) Physical access restrictions to server hardware, (4) Automatic updates of external components, (5) Encrypted backups, (6) Database encryption, (7) Penetration testing, (8) Intrusion detection system, (9) Firewalls and virus scanners

Table 31. Risk assessment (III)

Scenario	**Threat Event**	**Threat Sources**	**Vulnerabilities and Predisposing Conditions**	**LoT**	**LoI**	**Risk**	**Countermeasures (Elements of the Security Architecture)**
Web-based attack on data store(s)	Information Disclosure	External attacker, Malicious insider	Insufficient access control	MED	MED	MED	(1) Role-based access control, (2) Physical access restrictions to client/server hardware, (3) Automatic updates of external components, (4) Encrypted backups, (5) Database encryption, (6) Penetration testing, (7) Firewalls and virus scanners, (8) Two-factor authentication
Denial of Service of data stores	Denial of Service	External attacker	Insufficient protection of system or missing input validation/ sanitization, not enough resources	LOW	LOW	LOW	(1) Server hardening, (2) Input validation/sanitization practices, (3) IP-based filtering of requests, (4) Penetration testing
Released data is linkable to patients/ users/ researchers	Linkability, Information Disclosure, Identifiability	External attacker, Malicious insider	Released data contains information that can be linked to a patient/ user/ researcher.	HIGH	MED	MED	(1) Standard operation procedures (Data release), (2) Data Use Agreements, (3) Encryption of non-physically secure data
Patient does not know for what his/her data is used.	Content unawareness	Organizational failure	Patient was not informed well enough.	LOW	MED	LOW	(1) Standard operating procedures (Patient information), (2) Data Use Agreements, (3) Ethics committee review

Table 32. Risk assessment (IV)

Scenario	**Threat Event**	**Threat Sources**	**Vulnerabilities and Predisposing Conditions**	**LoT**	**LoI**	**Risk**	**Countermeasures (Elements of the Security Architecture)**
User does not know for what his/her data is used.	Content unawareness	Organizational failure	User was not informed well enough.	LOW	LOW	LOW	(1) Standard operating procedures (patient information), (2) Data Use Agreements
Consent does not cover how data are managed, processed and shared.	Policy and consent non-compliance	Organizational failure	None or insufficient consent management	MED	MED	MED	(1) Consent Management, (2) Ethics committee review
Social engineering attack on user	Information disclosure Identifiability	External attacker	Basic security unawareness of users or patients	LOW	HIGH	MED	(1) Username/Password policies, (2) User security and compliance trainings, (3) Two-factor authentication
Overcapacity failure of data stores	Tampering, Denial of service	External attacker	Missing Input validation/sanitization, Missing handling of overcapacity failures	LOW	MED	LOW	(1) Input validation/sanitization practices, Server hardening, (2) IP-based filtering of requests, (3) Backups/disaster recovery plan, (4) Redundant server hardware/raid, (5) Virtualization, (6) Intrusion detection system, (7) Penetration testing

Table 33. Risk assessment (V)

Scenario	Threat Event	Threat Sources	Vulnerabilities and Predisposing Conditions	LoT	LoI	Risk	Countermeasures (Elements of the Security Architecture)
Data entry and changes by physician or lab not logged	Repudiation	Systemic shortcoming	Weak logging, Missing audit trail	LOW	LOW	LOW	(1) Auditing and logging
Hardware theft, brute force	Linkability, Information Disclosure, Denial of service	External attacker	No or Insufficient hardware protection	LOW	MED	LOW	(1) Physical separation and spatial distribution of data, (2) Physical access restrictions to server hardware, (3) Server rooms linked to response center
Paper document compromised	Non-repudiation Linkability Detectability Information disclosure Identifiability	External attacker	Insecure storage or transfer	LOW	LOW	LOW	(1) Standard operating procedures (Paper documents)
Data changes in the data stores (not traced)	Repudiation	External attacker, Malicious insider	Weak logging, Missing audit trail, No or weak physical access control	LOW	MED	LOW	(1) Auditing and logging, (2) Physical access control

Table 34. Risk assessment (VI)

Scenario	Threat Event	Threat Sources	Vulnerabilities and Predisposing Conditions	LoT	LoI	Risk	Countermeasures (Elements of the Security Architecture)
Server room fire	Denial of service	External attacker, Technical failure	No or Insufficient fire protection	LOW	MED	LOW	(1) Secure server rooms including UPS and fire extinguisher, (2) Physical access restrictions to server hardware, (3) Server rooms linked to response center
Brute force/social engineering of admin password	Tampering, Linkability Detectability Information disclosure Identifiability, Denial of service	External attacker, Malicious insider	No admin password management	LOW	HIGH	MED	(1) Server and application admin password management, (2) Penetration testing (3) Two-factor authentication
Attacker tries to exploit vulnerability of external component	Tampering, Linkability Detectability Information disclosure Identifiability, Denial of service	External attacker, Malicious insider	No or insufficient protection of external components (cf. Table 12)	LOW	HIGH	MED	(1) Automatic updates of external components, (2) External security requirements (ESR-1 – ESR-26, cf. Table 16 and Table 17)

4.4 Discussion

4.4.1 Principal Results

Many of the countermeasures deployed and implemented in our systems are well-known and in widespread use. First, we apply hardware-level protection, including restricted access to hardware, secure server rooms with a UPS, and redundant server hardware. Second, we implement network-level measures, such as communication based on TLS with certificates and IP-based filtering of requests. On the host-level, we perform backups and maintain disaster recovery plans, deploy intrusion detection systems, firewalls, virus scanners, perform penetration testing and server hardening and use virtualization as well as automated server updates. On the application-level, our software uses common methods, such as limits for login attempts, automated logout after a certain time period, two-factor authentication, role-based access control, input sanitization (e.g. against SQL injection) and input validation. Additionally, our software implements various pseudonymization methods, as described previously. On the client-level, we employ account management policies and perform user trainings.

Additionally, there are some more-specific security measures implemented by our system. We have covered many of them in the previous sections: prevention of replay attacks on the token infrastructure (one-time access tokens), distributed non-delegated authentication where each component handles authentication and authorization autonomously (distributed authorization), an audit trail that keeps protocol of every data modification on each backend (audit trail) and the encryption of master data in the according backend (database encryption). Additionally, users from a specific participating site are only allowed to access data of patients recruited at their site (site-based view). This is implemented with the role-based access control mechanism.

The site-based view of data is resulting in logical separated but physically central identification service which needs special protection due to the sensitive data stored. We decided to introduce strong authentication to better secure user-accounts. The OTP-Tokens used for this step are autonomous, time-based and do not need any kind of connection to the system. They assignment of strong authentication to a user's account is optional and done centrally by an administrator. Attacks like Phishing or Brute-Force are mitigated by using OTP-Tokens. Furthermore, exponentially increased response time, in case of a failed login attempt, additionally mitigates those attacks. While strong authentication makes compromising an account using standard attacks like Brute-Force almost impossible, a successful attack on one user account would only affect data of one site. Since all data are pseudonymized compromising an entire subsystem would not lead a disclosure of all data. To link the data, an attacker would need to compromise at least two other subsystems which are spatially, organizationally separated from each other and partially secured by encryption.

Identifying patient data on paper, e.g., the informed consent, remains in control of the recruiting site. Access to information is depending on roles. While researchers can have multiple roles in a project, different roles of a single researcher imply possible conflict of interest; a potential misuse of data must be taken into consideration.

The effects of pseudonymization are twofold. Firstly, an attacker might only be able to access sensitive data, which makes it difficult to re-identify the data subject. Secondly, an attacker might only have access to identifying data, which makes it difficult to cause any harm. The relationships between the separated data are protected with *pseudonyms*, which are artificial identifiers for which the link to the corresponding entity is kept *confidential* [Itut10].

Therefore, the separation of powers and duties can prevent the misuse of organizational powers. The unauthorized escalation of privileges is an attack that can be performed from the in- and outside. Besides policy and consent non-compliance, the information disclosure of one or multiple data stores is a threat with high impact. If no separation of powers and duties realized, the likelihood of escalation of privileges attack increases. Organizational structures concerning supervision might conflict with project internal structures. It is obvious, that the RBAC must be in place following exactly the need-to-know-principle as well as principle of least privilege. This requires the possibility of precise adjustment of the rights for each role. Further data access restrictions like a site-based view are recommended as well and covered by the need-to-know principle. User account management policies are essential in this context.

The manipulation of client software systems is a serious threat that must be taken into consideration. The attack can be performed from an insider or an external attacker. It can lead to linkability, information disclosure and in the worst case to identifiability. The multitude of clients accessing the system cannot be completely controlled. In a clinical environment, client systems are protected with firewalls and virus scanners, software installation policies as well as physical access restrictions. However, one has to rely upon clinical and IT staff on each site to take precautionary measures concerning state-of-the-art security of hardware and software as well as passwords management. A successful web-based attack on a data store may lead to information disclosure. The attacker may be an external person or an insider. Insufficient logical access control mechanism can lead to a successful attack. The countermeasures that need to be in place are RBAC, physical access restrictions to client and server hardware, automatic updates of external components, encrypted backups, database encryption, firewalls and virus scanners. To identify vulnerabilities, penetration testing on a regular basis is recommended. The data for release has to be distinguished from data that is published. Data which is released for research is likely to contain many quasi identifiers. The risk of released data being linkable to a patient, a user or a researcher has a medium risk. Countermeasures for mitigation of linkability comprise encryption of the data for secure handover as well as data use agreements. In the latter, researchers agree to make no attempt of re-identification and to report conspicuities. Standard operating procedures for data release comprise transformation of data to comply with HIPAA standard. The threat of policy and consent non-compliance of managed, processed and shared data can results from none or insufficient consent management. This can be mitigated by regular consent management practices and ethics committee reviews. A successful social engineering attack on a user would have a high impact and could lead to information disclosure and/or identifiability, depending on the level of access. The basic security unawareness of users might be a vulnerability in this context. For mitigation we recommend username/password policies as well as user security and compliance trainings. A brute force attack on a server or application admin password would have a high impact. It could result from insufficient password management strategies. To mitigate the threat, we recommend two-factor authentication mechanisms, server and application password management as well as penetration testing. The threat of an attacker exploiting vulnerabilities of external components could also have a high impact. The insufficient protection of external components makes a system vulnerable to known exploits. It is recommended to perform automatic updates for external components and to orient towards the security and technical implementation guides (STIGS) for the concerning components. Furthermore, CERT-notifications from the Computer Emergency Readiness Team should also be considered.

4.4.2 Comparison with Prior Work

From a security and privacy perspective, current pseudonymization concepts should be based on risk and threat analyses. This may be the reason why multiple schemes have been proposed but international consensus is missing. Overviews have been provided by [Chur03], [AKRK13] and [NeKo09]; the schemes described differ in their requirements on application level as well as on data level. Some of these differences can be explained with the fact that the schemes have been developed for different use cases (e.g. for data warehouses [KSMM05] as compared to research networks [PDHG14]). But still, many of the inherent design decisions seem to be ad-hoc and lack thorough justification, which could have been provided by a risk and threat analysis. Some requirements can be well justified with general principles in IT security, e.g., the need-to-know principle and the principle of least privilege. Other methods specified by pseudonymization concepts, however, have a strong impact on system design but lack such justification. Among the important open questions are motivations for the application-level requirements *LSR-20* (client-side re-combination only) and *LSR-21* (confidentiality of internal identifiers) as well as the data-layer requirement *LSR-22* (two-tier pseudonymization).

As already noted, a thorough risk and threat analysis is needed to determine to which extent pseudonymity and related methods, such as multi-tier pseudonymity or client-side re-combination of data, offer protection against common security threats at which costs. This in turn requires an analysis of potential attack vectors, risks associated with common types of data, methods for quantifying re-identification risks, and a consideration of results from related research areas. An analysis of this kind would exceed the scope of this thesis. In this work, we do not focus on the methodical basis of pseudonymity, but on its implementation. Analogously to related work [AKRK13, DSQS12], we will therefore simply assume that implementing pseudonymity as currently conceptualized offers protection against information disclosure.

4.4.3 Limitations

We agree with Deng et al. [DWSP11] that privacy principles depend on security principles. When analyzing both together, conflicts might occur. Good examples are non-repudiation and plausible deniability, where preserving the one can negatively affect the other. The qualitative assessment of likelihood of threat, level of impact and overall risk is a first approach to a sound and comprehensive risk and threat analysis. It does not cover the complexity of possible re-identification attacks and has to be formalized with a model.

4.4.4 Conclusions

The current use of concepts has been critically reviewed by some authors, e.g., [KaAD04, NeHe11]. We observe a tendency to separate research data into repositories related to data relevance for functional units, e.g. clinical data, lab data, analysis data [DSQS12, HDRP10]. Hereby, only fragments of the data may be disclosed if a particular data pool is successfully attacked. Likewise, if an attacker is eavesdropping on a communication link, pseudonymity is a relevant countermeasure, as data is distributed over several servers and therefore exchanged via different channels [PKLK15].

Many (if not most) potentially identifying attributes are not collected for re-identifying data subjects during follow-ups but for characterizing the phenotype [LoDM10] or the genotype of individuals [GMGH13, HSRD08]. Distinguishability of both attribute types may be high, and for the important example of genomic data an extreme case, i.e. uniqueness of a sequence, is reached. Linkage to a person requires reference sets, which have different availability for different attributes. For genomic data, such references may be technically easy to obtain (e.g. by analyzing a hair of a known person), leading to a discussion on legal measures. Sensitive attributes which could be used for re-identification or which are even quasi-identifiers are not necessarily collected for (re-)identifying data subjects during follow-ups but for characterizing the phenotype [LoDM10] or (even) the genotype of individuals [GMGH13, HSRD08]. On the other side, the stringent separation of attributes considered “identifying” from the medical data does not make any difference between the potential of these attributes to be useful in re-identification attacks. It has been shown however that it is often specific subsets of such attributes that can cause privacy breaches [CiVF07, Swee00].

The main security risk remains the user (e.g. physician or researcher) with the device, its software and environment that is used to connect to the system. Due to decentralized and heterogeneous environment of clinical workstations malicious software can cause problems on client systems. To minimize these risks we recommend special trainings for physicians using the system and we are considering preconfigured devices for usage. This would partially eliminate the threat of malware on client systems. It would also offer the possibility for patients to enter data (e.g., quality of life data) at home. Although the exported data is in conformance with HIPAA limited data set, the risk of re-identification for patients contained in the exported dataset cannot be entirely excluded.

A proper risk assessment is the basis for identifying countermeasures for the biggest threats (i.e. threats most likely, and causing the most harm). The systematic measurement of risk is challenging and existing approaches address the problem of risks assessment for pseudonymization architectures only partially [PKLK15].

A comprehensive methodology for achieving pseudonymity in biomedical research is inevitable for fostering the development of consistent and secure solutions. We consider the steps proposed in this work the basis for such a model any beyond that, we hope that our results act as a stimulus for developing more elaborated systems with stringent privacy guarantees and consistent solutions to similar problems.

Bibliography

[00a]	International Cancer Genome Consortium: http://icgc.org/icgc/goals-structure-policies-guidelines/b-consortium-goals. Accessed 10 June 2018.
[00b]	BioMedBridges. http://www.biomedbridges.eu/. Accessed 10 June 2018
[00c]	The German National Cohort: http://www.nationale-kohorte.de/informationen_en.html (2015). Accessed 10 June 2018.
[00d]	AngularJS: https://angularjs.org (2015). Accessed 10 June 2018.
[00e]	BACKBONE.JS: http://backbonejs.org (2015). Accessed 10 June 2018.
[00f]	JGroups: http://www.jgroups.org (2015). Accessed 10 June 2018.
[00g]	OATH Standard: http://www.openauthentication.org (2015). Accessed 10 June 2018.
[00h]	Starlims: http://www.starlims.com/de-de/home (2015). Accessed 10 June 2018.
[00i]	secuTrial: http://www.secutrial.com (2015). Accessed 10 June 2018.
[13]	Shibboleth 3: a new identity platform. 2013. https://shibboleth.net/documents/business-case.pdf. Accessed 10 June 2018.
[ACCH13]	Ayday E, De Cristofaro E, Hubaux J-P, Tsudik, G. The chills and thrills of whole genome sequencing. IEEE Computer. 2013; doi:10.1109/MC.2013.333.
[ACKM04]	Alonso G, Casati F, Kuno H, Machiraju V. Web services: Concepts, architectures and applications (Data-centric systems and applications). Springer Berlin Heidelberg; 2004. p. 123-149. ISBN-10: 3642078885.
[AKRK13]	Aamot H, Kohl CD, Richter D, Knaup-Gregori P. Pseudonymization of patient identifiers for translational research. BMC Med Inform Decis Mak. 2013; 13(1):75 doi:10.1186/1472-6947-13-75.

[AnLo04]	Anderson A, Lockhart H. SAML 2.0 profile of XACML. OASIS Open 2004. http://docs.oasis-open.org/xacml/access_control-xacml-2.0-saml_profile-spec-cd-01.pdf. Accessed 10 June 2018.
[AnRo01]	Anderlink MR, Rothstein MA. Privacy and confidentiality of genetic information: what rules for the new science? Annual review of genomics and human genetics (2); 2001. p. 401-433. ISBN: 1527-8204.
[ApJo10]	Appari A, Johnson ME. Information security and privacy in healthcare: current state of research. Int J Internet Enterp Manag. 2010; 6(4):279-314; doi:10.1504/IJIEM.2010.035624.
[ASWS08]	Angelow A, Schmidt M, Weitmann K, Schwedler S, Vogt H, et al. Methods and implementation of a central biosample and data management in a three-centre clinical study. Computer methods and programs in biomedicine. 2008. 91(1):82-90. ISBN: 0169-2607.
[BaVA99]	Barry MJ, VanSwearingen JM, Albright AL. Reliability and responsiveness of the barry-albright dystonia scale. Dev Med Child Neurol. 1999; 41(6):404–411.
[BBHP15]	Bialke M, Bahls T, Havemann C, Piegsa J, Weitmann K, Wegner T, et al. MOSAIC – A modular approach to data management in epidemiological studies. Methods Inf Med. 2015; 54:364–371; doi:10.3414/ME14-01-0133.
[Beva97]	Bevan N. Usability issues in web site design. http://experiencelab.typepad.com/files/usability-issues-in-website-design-1.pdf (1999). Accessed 10 June 2018.
[BGLK12]	Büchner B, Gallenmüller C, Lautenschläger R, Kuhn KA, Wittig I, Schöls L et al. Das deutsche Netzwerk für mitochondriale Erkrankungen (mitoNET). Medizinische Genetik. 2012; 24(3):193-199; doi:10.1007/s11825-012-0338-8.
[BKGÜ05]	Brinkmann L, Klein A, Ganslandt T, Ückert F. Implementing a data safety and protection concept for a web-based exchange of variable medical image data. Int Congr Ser 1281. 2005:191-195; doi:10.1016/j.ics.2005.03.185.

[Chur03]	Churches T. A proposed architecture and method of operation for improving the protection of privacy and confidentiality in disease registers. BMC Medical Research Methodology. 2003. 3(1).
[CiVF07]	Ciriani V, Di Vimercati SDC, Foresti S, Jajodia S, Paraboschi S, et al. Fragmentation and encryption to enforce privacy in data storage. In: European Symposium on Research in Computer Sec. Springer, Berlin, Heidelberg. 2007. p. 171-186.
[ClMM03]	Claerhout B, De Moor GJE, De Meyer F. Secure communication and management of clinical and genomic data: the use of pseudonymisation as privacy enhancing technique. Stud Health Technol Inform. 2002; 95:170-175; doi: 10.3233/978-1-60750-939-4-170.
[COEC12]	Council of European Union: Directive of the European Parliament (draft replacement Directive 95/46/EC). In: Communication Bd. 10 (2012)
[Coun06]	Council of European Union: Recommendation Rec(2006) 4 of the Committee of Ministers to member states on research on biological materials of human origin. 958th meeting. 15 March 2006.
[CrHC08]	Crites S, Hsu F, Chen H. Omash: enabling secure web mashups via object abstractions. In Proc 15th ACM conf on Computer and communications Sec CCS. 2008. p. 99-108
[DaRK98]	Dadam P, Reichert M, Kuhn KA. Clinical workflows-the killer application for process-oriented information systems?. In: Abramowicz W, Orlowska ME, editors, BIS 2000, 4th Int Conf on Bus Inf Syst, Springer London. 2000. p. 36-59.
[DDGH10]	Dangl A, Demiroglu SY, Gaedcke J, Helbing K, Jo P, Rakebrandt F, et al. The IT-infrastructure of a biobank for an academic medical center. Stud Health Technol Inform 160 Pt 2. 2010:1334-8; doi:10.3233/978-1-60750-588-4-1334.
[DeSS11]	Demiroglu SY, Skrowny D, Schulze TG. Adaption of the identity management regarding new requirements of a long-term psychosis biobank. In: Moen A, Andersen SK, Aarts J, Hurlen P, editors. In Proc 23rd Int Conf European Federation Med Inform. MIE 2011. 2011:1–3.

[DSQS12]	Demiroglu SY, Skrowny D, Quade M, Schwanke J, Budde M, Gullatz V et al. Managing sensitive phenotypic data and biomaterial in large-scale collaborative psychiatric genetic research projects: practical considerations. Mol Psychiatry. 2012; 17(12):1180–1185; doi:10.1038/mp.2012.11.
[DWSP11]	Deng M, Wuyts K, Scandariato R, Preneel B, Joosen W. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering. 2011. 16(1):3-32.
[EmAr13]	El Emam et al. Anonymizing health data: case studies and methods to get you started. O'Reilly Media Inc. Hrsg. 2013. ISBN 978-1-4493-6307-9
[Euro01]	European Commission: EU Directive 2001/20/EC. Official Journal of the European Communities. 2001:1–15. ISBN 9789279122552.
[Euro05]	European Commission: EU Directive 2005/28/EC. Official Journal of the European Union. 2005. L 91(13):13-19.
[Euro08a]	European Commission: Annex 11 - Computerized Systems. EudraLex - Guidelines for good manufacturing practices for medicinal products for human and veterinary use. Communication. 2008. ISBN 1453384154.
[Euro08b]	European Clinical Research Infrastructure Network: GCP-compliant data management in multinational clinical trials. 2008(0):1-49.
[Euro12]	European Commission: FP7-HEALTH - FP7 specific programme 'cooperation' - research theme: 'health'. http://cordis.europa.eu/programme/rcn/852_en.html (2007). Accessed 10 June 2018.
[EWAG07]	Eggert K, Wüllner U, Antony G, Gasser T, Janetzky B, Klein C, et al. Data protection in biomaterial banks for parkinson’s disease research: the model of GEPARD (gene bank parkinson's disease germany). Mov Disord. 2007; 22(5):611-318; doi:10.1002/mds.21331.
[FaPo05]	Faldum A, Pommerening K. An optimal code for patient identifiers. Computer methods and programs in biomedicine. 2005. 79(1):81-88.
[Fda00]	FDA: Guidance for Industry Computerized Systems Used in Clinical Trials. 2004.

[Fede09]	Federal Data Protection Act in the version promulgated on 14 January 2003 (Federal Law Gazette I p. 66), as most recently amended by Article 1 of the Act of 14 August 2009 (Federal Law Gazette I p. 2814). 2009.
[Fede17]	Federal Republic of Germany: Gesetz zur Anpassung des Datenschutzrechts an die Verordnung (EU) 2016 / 679 und zur Umsetzung der Richtlinie (EU) 2016 / 680 (Datenschutz-Anpassungs- und -Umsetzungsgesetz EU – DSAnpUG-EU). Bundesanzeiger. 2017. 44:2097–2132
[Fkoh10]	Kohlmayer F, Lautenschläger R, Wurst SHR, Klopstock T, Prokisch H, Meitinger T, et al. Konzept für ein deutschlandweites Krankheitsnetz am Beispiel von mitoREGISTER. GI Jahrestagung. 2010:746–751.
[FlKe02]	Florence K, Kelly E. 21 CFR Part 11 Electronic Records. Electronic Signatures. 2002:1–11.
[Garf15]	Garfinkel SL. De-identification of personal information. National Institute of Standards and Technology. 2015; doi:10.6028/NIST.IR.8053.
[Gcpi07]	GCP Inspections Working Group: EMEA Reflection paper on expectations for electronic source documents used in clinical. 2007:1-10.
[GMGH13]	Gymrek M, McGuire AL, Golan D, Halperin E, Erlich Y. Identifying personal genomes by surname inference. Science. 2013. 339(6117):321-324.
[GoNT08]	Goldberg SI, Niemierko A, Turchin A. Analysis of data errors in clinical research databases. AMIA Annu Symp Proc. 2008:242–246.
[HaGS03]	Hakonarson H, Gulcher JR, Stefansson K. deCODE genetics, Inc. Pharmacogenomics. 2003; 4:209–215.
[HDRP10]	Helbing K, Demiroglu SY, Rakebrandt F, Pommerening K, Rienhoff O, Sax U. A data protection scheme for medical research networks. Methods Inf Med. 2010; 49(6):601-607; doi:10.3414/ME09-02-0058.
[HeKN11]	Heurix J, Karlinger M, Neubauer T. Pseudonymization with metadata encryption for privacy-preserving searchable documents. In: Proc Annu Hawaii Int Conf Syst Sci. HICSS 2012. 2012:3011-3020; doi:10.1109/HICSS.2012.491.

[Hl00]	The HL7 CCOW Standard: http://www.hl7.com.au/CCOW.htm (2006). Accessed 10 June 2018.
[HoLi06]	M. Howard und S. Lipner. The security development lifecycle: SDL, a process for developing demonstrably more secure software. Microsoft Press; 2006. ISBN-10: 0735622140
[HSRD08]	Homer N, Szelinger S, Redman M, Duggan D, Tembe W. et al. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. PLoS genetics. 2008. 4(8):e1000167; doi:10.1371/journal.pgen.1000167.
[Iaco07]	Lo Iacono L. Multi-centric universal pseudonymisation for secondary use of the EHR. Stud Health Technol Inform. 2007; 126:239–247.
[Ietf81]	University of Southern California. RFC 791: Darpa internet program protocol specification. 1981. https://tools.ietf.org/html/rfc791. Accessed 10 June 2018.
[Inst00]	DSLib: http://www.unimedizin-mainz.de/imbei/informatik/opensource/dslib.html (2015). Accessed 10 June 2018.
[Inte00]	International Organization for Standardization (ISO): Health informatics - pseudonymization. ISO/TS 25237:2008(E). 2008.
[Isoi09]	International Organization for Standardization (ISO): Information technology - security techniques - information security management systems - overview and vocabulary. ISO/IEC 27000:2009(E). 2009.
[Iso16]	International Organization For Standardization (ISO): Information security - security techniques - information security management systems - Requirements. ISO 27001:2016(E). 2016.
[Itut10]	ITU-T: X.1252 - Baseline identity management terms and definitions. In: ITU-T X-Series Recommendations Data Networks, Open System Communications and Sec. 2010.
[IvGr96]	Iversen K, Grøtan T. Socio-technical aspects of the use of health related personal information for management and research. Int J Biomed Comput. 1996; 43(1):83–91.

[JaBo06]	Jackson C, Bortz A, Boneh D, Mitchell JC. Protecting browser state from web privacy attacks. In: Proc Int Conf World Wide Web. 2006:737-744; doi:10.1145/1135777.1135884.
[JAHC09]	Jin J, Ahn G-J, Hu H, Covington MJ, Zhang X: Patient-centric authorization framework for sharing electronic health records. In Proc 14th ACM Symp Access Control Model Technol. 2009; 125-134; doi 10.1145/1542207.1542228.
[JaWa07]	Jackson C, Wang HJ. Subspace: Secure cross-domain communication for web mashups. In: Proc Int Conf World Wide Web. 2007:611-620; doi:10.1145/1242572.1242655.
[Jone11]	Jones MB. The emerging JSON-based identity protocol suite. W3C workshop on identity in the browser. 2011:1–3.
[Jour16]	European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). In: Official J of the European Union. 2016. L 119.
[KaAD04]	Kalam E, Abou A, Deswarte Y, Trouessin G, Cordonnier E. A generic approach for healthcare data anonymization. In Proceedings of the 2004 ACM workshop on Privacy in the electronic society. 2004. p. 31-32; doi:10.1145/1029179.1029188.
[KBSC08]	De Keukelaere F, Bhola S, Steiner M, Chari S, Yoshihama S. Smash: secure component model for cross-domain mashups on unmodified browsers. In Proceedings of the 17th Int Conf on World Wide Web. 2008. p. 535-544.
[KLKB12]	Kalman B, Lautenschlaeger R, Kohlmayer F, Büchner B, Kmiec T, Klopstock T et al. An Int registry for neurodegeneration with brain iron accumulation. Orphanet J Rare Dis. 2012; 7:66; doi:10.1186/1750-1172-7-66.
[KSMM05]	Kalra D, Singleton P, Milan J, MacKay J, Detmer D, Rector A, et al. Security and confidentiality approach for the clinical e-science framework (CLEF). Methods Inf Med. 2005; doi:10.1267/METH05020193.
[LaBÜ15]	Lablans, M, Borg, A, Ückert, F. A RESTful interface to pseudonymization services in modern web applications. BMC Med Inform Decis Mak. 2015; 15(1):2; doi:10.1186/s12911-014-0123-5.

[LiMC15]	Liu V, Musen MA, Chou T. Data breaches of protected health information in the United States. Jama. 2015. 313(14):1471-1473; doi: 10.1001/jama.2015.2252
[LKPK15]	Lautenschläger R, Kohlmayer F, Prasser F, Kuhn KA. A generic solution for web-based management of pseudonymized data. BMC medical Inform and decision making. 2015. 15(1):100; doi:10.1186/s12911-015-0222-y.
[LoDM10]	Loukides G, Denny JC, Malin B. The disclosure of diagnosis codes can breach research participants' privacy. J Am Med Inform Assoc. 2010; 17(3):322-327; doi:10.1136/jamia.2009.002725
[Lowr03]	Lowrance W. Learning from experience: privacy and the secondary use of data in health research. J Health Serv Res Policy 2003; 8 Suppl 1: 2-7; doi:10.1258/135581903766468800.
[Mali05]	Malin B. An evaluation of the current state of genomic data privacy protection technology and a roadmap for the future. J Am Med Inform Assoc. 2005. 12:28–34.
[MaSc12]	Majchrzak T, Schmitt O. Improving epidemiology research with patient registries based on advanced web technology. In: Proc Int Conf Info Sys Crisis Response Management. 2012:1–5.
[McGK10]	McCallister E, Grance T, Scarfone KA. Guide to protecting the confidentiality of personally identifiable information (PII). Special Publication (NIST SP)-800-122. 2010. ISBN: 9781437934885.
[MeMR08]	De Meyer F, De Moor G, Reed-Fourquet L. Privacy Protection through pseudonymisation in eHealth. Studies in health technology and Inform. 2008. 141:111-118.
[MFJK07]	Mailman MD, Feolo M, Jin Y, Kimura M, Tryka K et al. The NCBI dbGaP database of genotypes and phenotypes. Nature genetics. 2007. 39(10):1181-1186; doi:10.1038/ng1007-1181.
[MHLO14]	Swiderski F, Snyder W. Threat modeling. Microsoft Press. 2004. ISBN-10: 0735619913.
[MLWÜ14]	Muscholl M, Lablans M, Wagner TO, Ückert F. OSSE: open source registry software solution. Orphanet J Rare Dis. 2014; 9 Suppl 1; doi:10.1186/1750-1172-9-S1-O9.

[MoCM03]	De Moor GJE, Claerhout B, De Meyer F. Privacy enhancing techniques: the key to secure communication and management of clinical and genomic data. Methods Inf Med. 2003; 42(2):148-153; doi:10.1267/METH03020148.
[MOFH12]	Meyer J, Ostrzinski S, Fredrich D, Havemann C, Krafczyk J, Hoffmann W: Efficient data management in a large-scale epidemiology research project. Comput Methods Programs Biomed. 2012; 107(3):425-435; doi:10.1016/j.cmpb.2010.12.016.
[Msdn11]	Meier JD, Mackman A, Dunner M, Vasireddy S, Escamilla R et al. Improving web application security: threats and countermeasures. Microsoft Corporation. 2003. URL: http://212.200.39.245:81/Crna%20Rupa/Arhiva/2009-2010/FIM/ZIS/Literatura/How%20To%20Use%20IPSec%20for%20Filtering%20Ports%20and%20Authentication.doc. Accessed 10 June 2018.
[MuLÜ14]	Muscholl M, Lablans M, Ückert F. OSSE: open source registry system for rare diseases in the EU (executive summary). 2014. https://fileshare.zdv.uni-mainz.de/yaD2X7kLbulPL9iofy-tcA.file. Accessed 10 June 2018.
[NeHe11]	Neubauer T, Heurix J. A methodology for the pseudonymization of medical data. Int J Med Inform. 2011; 80(3):190-204; doi:10.1016/j.ijmedinf.2010.10.016.
[NeKo05]	Neuman C, Kohl J. RFC 4120: The Kerberos network authentication service (V5). 2005. http://www.ietf.org/rfc/rfc4120.txt. Accessed 10 June 2018.
[NeKo09]	Neubauer T, Kolb M. An evaluation of technologies for the pseudonymization of medical data. In: Computer and Information Science. Springer Berlin Heidelberg. 2009:47-60; doi:10.1007/978-3-642-01209-9_5.
[Niel12]	Christensen H, Nielsen JS, Sørensen KM, Melbye M, Brandslund I. New national biobank of The Danish Center for Strategic Research on Type 2 Diabetes (DD2). Clinical epidemiology. 2012. 4:37-42; doi:10.2147/CLEP.S33042.
[Nist06]	Ross RS, Katzke SW, Johnson LA. Minimum security requirements for federal information and information systems. Federal Inf. Process. Stds. 2006; doi:10.6028/NIST.FIPS.200
[Nist12]	Stoneburner G, Goguen AY, Feringa A. Sp 800-30. Risk management guide for information technology systems. National Institute of Standards Technology. 2002.

[NoLL07]	Noumeir R, Lemay A, Lina JM. Pseudonymization of radiology data for research purposes. J Digit Imaging. 2007; 20(3):284-295; doi:10.1007/s10278-006-1051-4.
[OKCL11]	Ohmann C, Kuchinke W, Canham S, Lauritsen J, Salas N, Schade-Brittinger C, et al. Standard requirements for GCP-compliant data management in multinational clinical trials. Trials 12.1. 2011; 12:85; doi:10.1186/1745-6215-12-85.
[OTBD15]	Van Ommen GJB, Törnwall O, Bréchot C, Dagher G, Galli J et al. BBMRI-ERIC as a resource for pharmaceutical and life science industries: the development of biobank-based Expert Centres. European J of Human Genetics. 2015. 23(7):893-900. doi:10.1038/ejhg.2014.235
[PDHG14]	Pommerening K, Drepper J, Helbing K, Ganslandt T. Leitfaden zum Datenschutz in medizinischen Forschungsprojekten. 1st ed. MWV; 2014. ISBN-10: 3954661233.
[PeOm13]	Perola M, Van Ommen GJ. BBMRI-LPC - A four-year project to help scientists to have better access to large European studies on health (2013). URL: http://old.bbmri-eric.eu/documents/10181/68479/BBMRI-LPC_Press+release_ENGLISH_final.pdf/97826f5b-87bd-4026-a5f0-36ce2fd59afd?version=1.0. Accessed 10 June 2018.
[PfHa10]	Pfitzmann A, Hansen M. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. 2010. URL:http://www.maroki.de/pub/dphistory/2010_Anon_Terminology_v0.34.pdf. Accessed 10 June 2018.
[Phar07]	Pharmaceutical Inspection Co-Operation Scheme (PIC/S): Good Practices for Computerised Systems in Regulated "Gxp" Environments. Pic/S. 2007(39):1-54.
[PKLK15]	Prasser F, Kohlamyer F, Lautenschläger R, Kuhn KA. Pseudonymity in Biomedical Research: A Model (Internal Working Paper). Institut für medizinische Statistik und Epidemiologie. Klinikum rechts der Isar. 2015.

[PRDS05]	Pommerening K, Reng M, Debold P, Semler S. Pseudonymisierung in der medizinischen Forschung - das generische TMF-Datenschutzkonzept. Pseudonymization in medical research - the generic data protection concept of the TMF. In: Medical Inform. 2005. 1(3):1–6.
[PrRi11]	Prokosch HU, Ries M, Beyer A, Schwenk M, Seggewies et al. IT infrastructure components to support clinical care and translational research projects in a comprehensive cancer center. In MIE. 2011. 169:892-896.
[PSMS08]	Pommerening K, Sax U, Müller T, Speer R, Ganslandt T, Drepper J et al. Integrating eHealth and medical research: The TMF data protection scheme. In: Blobel B, Pharow P, Zvarova J, Lopez D, editors. eHealth: Combining health telematics, telemedicine, biomedical engineering and bioinformatics to the edge. Akademische Verlagsgesellschaft Aka GmbH: Berlin, 2008:5-10.
[Publ00]	P3G: Public population project in genomics and society. http://p3g.org (2015). Accessed 10 June 2018.
[RDDP12]	De Ryck P, Decat M, Desmet L, Piessens F, Joosen W. Security of web mashups: a survey. In: Proc Nord Conf Sec IT Syst. 2012:223-238; doi:10.1007/978-3-642-27937-9_16.
[Repu03]	Republic of Italy. Personal data protection code. legislative decree No. 196, (196), 1–186. 2003.
[RiGN08]	Riedl B, Grascher V, Neubauer T. A secure e-health architecture based on the appliance of pseudonymization. J Software. 2008; 3(2):23-32; doi:10.4304/jsw.3.2.23-32.
[RKBS08]	Robinson PN, Köhler S, Bauer S, Seelow D, Horn D, Mundlos S. The human phenotype ontology: a tool for annotating and analyzing human hereditary disease. Am J Hum Genet. 2008; 83(5):610-615: doi:10.1016/j.ajhg.2008.09.017.
[Room10]	Room A. Dictionary of Pseudonyms: 13,000 assumed names and their origins. McFarland. 2010. ISBN: 9780786457632.
[Savo96]	Savoy-Lewis A. Health Insurance Portability and Accountability Act of 1996: a tempered victory. In: The J of law, medicine & ethics : a J of the American Society of Law, Medicine & Ethics. 1996. 24:380–385.

[Schu99]	Schulmeister L. Privacy and confidentiality. Clinical J of oncology nursing. 1999. 3(1):34-35.
[ScPE12]	Schaefer AM, Phoenix C, Elson JL. Mitochondrial disease in adults: a scale to monitor progression and treatment mitochondrial disease in adults. Neurology. 2012; 66(12):1932–4.
[Shir07]	Shirey RW. Internet security glossary, version 2. In: Request for Comments (RFC 4949). 2007.
[SMBB06]	Schmitz-Hübsch T, Du Montcel ST, Baliko L, Berciano J, Boesch S, Depondt C et al. Scale for the assessment and rating of ataxia: development of a new clinical scale. Neurology. 2006; 66(11):1717–1720.
[Soci05]	Society For Clinical Data Management: Good Clinical Data Management Practices. 2005. p.141.
[SoSh13]	Son S, Shmatikov V. The postman always rings twice: attacking and defending postMessage in HTML5 websites. In: ISOC Network and Distributed System Sec Symposium, NDSS 2013. 2013.
[SpUU09]	Spitzer M, Ullrich T, Ueckert F. Securing a web-based teleradiology platform according to German law and “Best Practices”. Stud Health Technol Inform. 2009; 150:730–734.
[Swee00]	Sweeney L. Simple demographics often identify people uniquely. Health (San Francisco). 2000. 671:1-34.
[Swee02]	Sweeney L. k-anonymity: A model for protecting privacy. Int J of Uncertainty, Fuzziness and Knowledge-Based Systems. 2002. 10(05):557-570.
[Syst10]	Schaeffer R. National information assurance (ia) glossary. CNSS Secretary NSA, Ft. Meade. 2010.
[Ukbi07]	UK Biobank: UK biobank ethics and governance framework (version 3.0). 2007. https://www.ukbiobank.ac.uk/wp-content/uploads/2011/05/EGF20082.pdf. Accessed 10 June 2018.

[Unif15]	Unified Compliance Framework: (STIG) Viewer. URL:https://www.stigviewer.com/stigs. Accessed 10 June 2018.
[Usde13]	U.S. Department of Health and Human Services Office for Civil Rights: HIPAA administrative simplification regulation, 45 CFR Parts 160, 162, and 164. 2013.
[Vie00]	Le Vie DS, Donald S. Understanding data flow diagrams. In Ann Conf - Society for Technical Communication. 2000. 47:396-401.
[WiJo91]	Wieringa R, De Jonge W. The identification of objects and roles-object identifiers revisited. In: Technical Report IR-267. 1991. p.1–16.
[WKWS11]	Wichmann HE, Kuhn KA, Waldenberger M, Schmelcher D, Schuffenhauer S, Meitinger T, et al. Comprehensive catalog of european biobanks. Nat Biotechnol. 2011; 29:795–7; doi:10.1038/nbt.1958.
[WyMi03]	Wylie JE, Mineau GP. Biomedical databases: protecting privacy and promoting research. Trends Biotechnol. 2003; 21(3):113-116; doi:10.1016/S0167-7799(02)00039-2.

Definitions

We will subsume the necessary definitions in this paragraph.

Threat	“[a]ny circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service. […]” [Nist06].
Attack	“Any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself” [Syst10].
Vulnerability	“Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source” [Syst10].
Risk	*“[…] the potential impact of a threat and the likelihood of that threat occurring.*” [Nist06]. We note that to measure risk is often a significant challenge, and different methodologies exist. In practice, risks prevented and risks remaining need to be contrasted with the costs of measures, and a balance should be sought [Nist12].
Privacy	“*the right of […] a person […] to determine the degree to which it [...] is willing to share its personal information with others*” [Shir07].
Personal information	is “[…] information […] that could cause harm or pain to that person if disclosed to unauthorized parties […]” [Shir07]. Examples include: *“Any information a) that identifies or can be used to identify, contact, or locate the person to whom such information pertains; b) from which identification or contact information of an individual person can be derived; or c) that is or can be linked to a natural person directly or indirectly*“ [Itut10].

Data subject	"[…] an identified or identifiable natural person […] who can be identified, directly or indirectly […]"[Jour16]..
Identification	is the process of *recognizing* an entity by a subset of its *characterizing attributes* [Itut10]. The subset is called *identifier* [Itut10] and the characterizing attributes that can cause harm are called *sensitive attributes*. Recognizing an entity also implies *distinguishing* it from others [Itut10].
Indirectly identifying data	data that can identify a single person only when used together with other indirectly identifying data [Inte00].
Attack	Any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself [Syst10].
Countermeasure	"Actions, devices, procedures, or techniques that meet or oppose (i.e., counters) a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken" [Syst10].
Privacy	*"the right of […] a person […] to determine the degree to which it [...] is willing to share its personal information with others"* [Shir07]. This requires that individuals trust all involved parties that their data is adequately protected and only shared and processed in a pre-specified manner [Schu99]. We will use this definition, without covering the complete legal and social/societal perspectives. For a broader view, we refer to [AnRo01], which clarifies that the concept "privacy" is not only related to data or information access, but also to access to persons and personal spaces.
Confidentiality	"property that information is not made available or disclosed to unauthorized individuals, entities, or processes" [Isoi09]. If data about individuals is collected and privacy has to be guaranteed, this requires ensuring confidentiality.

Entity	„Something that has separate and distinct existence […]“ [Itut10].
Attribute	„Information bound to an entity that specifies a characteristic of the entity“ [Itut10].
Identification	“The process of recognizing an entity by contextual characteristics” [Itut10]. Recognizing an entity also implies *distinguishing* it from others [Itut10].
Identifier	“One or more attributes used to identify an entity within a context” [Itut10]. Identifiers can be *open* or *secret* [Itut10]. The latter means that the binding to the corresponding entity is kept confidential in a specific context. Moreover, the *visibility* of identifiers can differ between contexts. Keeping it *invisible* means that it will not be shown to users [WiJo91].
Pseudonym	“An identifier whose binding to an entity is not known or is known to only a limited extent, within the context in which it is used” [Itut10]. We note that this means that a pseudonym is a secret identifier. Further definitions exist, e.g., [Inte00].

List of Abbreviations

AJAX	Asynchronous JavaScript and XML
CCOW	Clinical context object workgroup
CDISC	Clinical Data Interchange Standards Consortium
CDMS	Clinical Data Management System
CORS	Cross-Origin resource sharing
CRF	Case Report Form
CRUD	create read update and delete
DBMS	Database Management System
DoS	Denial of Service
DUA	Data Use Agreement
EC	Ethic Committee
eCRF	Electronic case report form
EDC	Electronic data capture
EHR	Electronic health record
ELSI	Ethical, Legal, and Social Implications
FP7	Seventh framework programme
GCP	Good Clinical Practice
HIPAA	Health Insurance Portability and Accountability Act
HL7	Health Level Seven
HTML	Hypertext Markup Language
HTTP	Hypertext transfer protocol

IFrame	Inline frame
IOI	Item of interest
IRB	Institutional review board
ISO	International Organization for Standardization
JS	JavaScript
JSON	JavaScript object notation
JSONP	JSON with padding
JWE	JSON web encryption
JWK	JSON web keys
JWS	JSON web signatures
JWT	JSON web tokens
LINDDUN	Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Content Unawareness, Policy and consent non-compliance
MVW	Model-View-Whatever
NIST	National Institute of Standards and Technology
OATH	Initiative for open authentication
OSSE	Open source registry system for rare diseases in the EU
OTP	One-Time-Password
RBAC	Role-based access control
RDBMS	relational database management system
REST	Representational state transfer
SAML	Security assertion markup language
SDL	Secure Development Lifecycle

SOP	Same-Origin-Policy
SPA	Single-page application
SSL	Secure Socket Layer
SSO	Single sign-on
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, Denial of service, Elevation of privilege
TLS	Transport layer security
TTP	Trusted third party
UPS	Uninterruptible power supply
URL	Uniform resource locator
USB	Universal Serial Bus
W3C	World Wide Web Consortium
Web API	Web application programming interface
WS	Web service
XACML	eXtensible access control markup language
XHR	XMLHttpRequest
XML	eXtensible Markup Language