PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI - University of Trento

Methods, Policies and Technologies for Compliance-Aware Management of Electronic Health Records

Jovan Stevovic

Advisor:

Prof. Fabio Casati

Università degli Studi di Trento

Co-Advisor:

Dr. Jun Li

HP Labs

April 2014

Abstract

Medical record sharing across healthcare organisations is fundamental for improving quality of care and reducing assistance costs. However, healthcare organisations are still struggling in building cross-organisation data sharing solutions due to strict data protection regulations that varies across states and regions, availability of a variety of technical standards for medical record sharing, and differences among organisations' IT infrastructures that have been built over the years to satisfy organisations' specific needs and requirements.

This thesis reports our findings based on various research and industrial projects aiming at connecting healthcare organisations. The primary contributions of this dissertation are:

- **A methodology and an execution environment to define and execute cross- organisation data sharing processes in compliance with both data protection regulations and organisations' requirements.** The methodology consists of multiple steps that start with the extraction of compliance requirements from regulations and gathering of business requirements from the involved stakeholders, and end with the definition of data sharing processes and policies to satisfy the collected requirements. The modelling framework that supports the methodology provides to users the modelling tools and guidelines to define the business processes and policies for sharing privacy-sensitive data. The execution framework maps the business processes into actionable operations to manage privacy-sensitive data and data protection policies.

- **An event-driven service integration approach to support cross-organisation data sharing.** The integration approach focuses on identifying data dependencies among institutions (i.e., data they produce, consume and would like to exchange) in form of events rather than analysing internal data structures. To support this approach, we propose a privacy-aware event-driven data-sharing protocol and a system architecture based on combination of Service Oriented (SOA) and Event Driven (EDA) architectural patterns. The data-sharing protocol and the underlying fine-grained access control policies provide control on the access and dissemination of sensitive information among the involved organisations.

- **A set of algorithms to detect and to prevent access control policy violations in data integration caused by the presence of functional dependencies.** In data integration typically each source specifies its local access control policies and cannot anticipate the functional dependencies among sets of attributes (or any other type of data inference) that can arise when data is integrated. Functional dependencies can allow malicious users to obtain prohibited information by linking multiple queries and thus violating the local policies. To solve such issues, we propose algorithms to identify the sets of queries that can lead to such privacy violations. We then propose algorithms to identify additional policies that are able to prevent the identified queries from completion and thus prevent policy violations.

We show how the proposed solutions have been applied in practice in building Electronic Health Record and Business Intelligence systems that involve cross-organisation sharing of privacy-sensitive data. The thesis reports also the validations of the proposed technologies with end-users and privacy experts, and the lessons learned after deploying an instance of the developed system in a multi-organisation scenario in Trentino, Italy.

Keywords
Electronic Health Records, regulatory compliance, cross-organization data sharing, business process modelling and execution, data sharing protocols, access control policies.

Посвећено мојој породици.
To my family.

Acknowledgements

I would like to express my gratitude to the countless brilliant people I met during this journey. Some of them taught me the fundamentals of research and helped me push on, but most of all, each one of them inspired me by exemplifying the makings of a great person.

First off, I want to thank Giampaolo Armellin and Fabio Casati. They had the brilliant idea for me to start a PhD. They created the right environment, provided guidance, and believed in my potential. Through their support I have come to learn earning a PhD is not only about (potentially) pushing the boundaries of human knowledge about a "tech topic". It comes with life lessons I can carry through the rest of my life.

My deepest gratitude goes also to Jun Li who, with the collaboration of Hamid Motahari and other brilliant researchers from the Service Automation and Integration Lab at HP Labs, helped me toward pursuing the PhD. He taught me how to do research and how to present research ideas.

Thanks to Claudio Bartolini who hosted a memorable experience at HP Labs. I had the opportunity to exchange ideas not only about informatics, but also important questions to which the answer is usually "42" [3].

I would like to thank Annamaria Chiasera for her support and motivation. I spent four memorable years with her and Dario, Tefo, Cristina, Tao, and all the other members of our team at Centro Ricerche GPI, under the supervision of Giampaolo. Besides the technical skills, what characterizes this group is an infinite amount of humanity and ethics.

Thanks to Karoline Beronius and the whole Shifo team for showing me how a small group of highly motivated people paired with a great vision can indeed make a difference. Inspiring.

It was a pleasure also to meet Marco Battisti. He showed me a different world and culture, and gave a practical example about ethics.

I am grateful to Bilal Farraj and Alessio Giori, two great interns who have contributed to the work presented in this dissertation.

I want to thank also all my friends too. Without their cheering (mainly at the pubs), I would have never overcome the many difficulties the PhD brings.

Finally, my deepest gratitude goes to my family. They continue showing me everyday the meaning of sacrifice, perseverance and life values.

И на крају, највише се захваљујем мојој породици. Они ми свакодневно показју шта значи пожртвованост, упорност и шта су праве животне вредности.

Contents

Chapter 1

Introduction

1.1 Motivation

Interoperable Electronic Health Record (EHR) systems can improve quality of care, reduce assistance costs and timely share data for clinical research and business intelligence over cross-organisation assistance processes [89, 145, 166]. Due to these benefits, the development of EHR systems has been identified as one of the strategic initiatives by many governments [67, 123, 174]. Although these initiatives have resulted into the development of different projects with promising outcomes [36, 67, 133, 136, 143], the current level of EHR adoption is still partial and evolves slowly [63].

Although the reasons for such partial success are multiple and involve for example human factors (e.g. resistance to change) and economical motivations (e.g. impact on existing IT systems), the technological issues have still a big impact [63, 145]. Therefore, a question that can arise is:

“Why is data sharing in healthcare more challenging from technological point of view than in other sectors like airline or banking? Why interoperability related challenges in airline and banking have been solved many years ago?"

The answer to this question sits in the complexity of the healthcare sector, which requires ad-hoc approaches and technologies for each integration scenario. While working on industrial and research projects in healthcare, we were faced with such complexity and observed that developing the technology to support medical record sharing involves challenges in the following aspects:

- **legal** aspects due to highly privacy-sensitive data content, different purposes of use of data, and strict data protection regulations defined at different levels. Due to the importance of privacy in healthcare, those regulations are defined also exclusively for medical record sharing [94] to identify data owners and entities responsible for managing and sharing data. Unlike banking data that contains only financial information, medical records contain much border personal information that includes patients' health, social and financial status. Therefore in healthcare, it is challenging to extract policies from regulations (usually requiring the involvement of actors having appropriate background and expertise), identify appropriate software development methodology to achieve compliance while sharing data, and prove to Privacy Guarantors, Compliance Officers and people with little technical background, that the developed solutions are compliant with regulations and policies.

- **technical** aspects due to existence of a variety of competing standards to deal with data semantics [20, 60, 161, 170], data sharing protocols [93, 140, 169] and system architectures [93, 143, 133, 169]. Furthermore, these standards are still evolving and there are still ongoing research efforts in identifying even more sophisticated solutions for the creation of interoperable EHR systems [87]. Therefore, for an organisation to join the EHR programs can be challenging and costly, since the organization would need to adapt its environment to these new standards. This obstacle becomes much more relevant for smaller institutions having less expertise and availability of resources.

- **organizational** aspects due to the intrinsic complexity of the healthcare assistance services which are delivered jointly by both public and private institutions of different sizes (e.g., large hospitals and small clinic offices) and focused on providing specific aids such as hospitalization, social assistance or assistance at home. By their nature, these services rely heavily on knowledge and experience of single caregivers. To satisfy their needs, institutions have developed over the years different IT systems specifically tight to best practices and based on how they have interpreted regulatory policies. To ensure their participation to EHR programs, and at the same time to preserve their working practices and existing information systems, there is the need to develop data sharing solutions that are as less intrusive as possible. As a consequence, it is difficult to develop a "one size fits all" solution.

These aspects and related challenges affect any research or industrial project that aims at developing solutions to enable cross-organisational data sharing in healthcare. For each project they can have different importance and impact on the resulting technology and their combination makes the development of data sharing in healthcare an extremely challenging task.

In the following sections, we report three research threads that span across these three aspects. For each of them we report the research challenges, our approach toward the solutions, the research contributions and the lessons learned on tackling it.

1.2 Research Challenges and Contributions

The overall aim of the work presented in this dissertation is at enabling easy, safe and regulatory compliant data sharing among healthcare organisations, and to facilitate the creation of interoperable EHR systems.

The thesis reports the identified challenges and research contributions within three different research threads. We start by analysing the challenges related to regulatory compliance in cross-organisation and cross-regulation data sharing, with the focus on developing tools to establish the links between regulatory policies and operations on data. We then focus on the required technology to support cross-organisation service integration. We describe a data sharing protocol and privacy policies and how they can achieve privacy-aware data sharing. Finally, we analyse the research challenges related to access control policy violation detection and we propose an approach to prevent such violations.

1.2.1 Compliance-Aware Medical Record Sharing

Data protection regulations define rules on medical records management at different levels, starting from unions/federations [69, 76] to individual countries [83, 132], internal regions or municipalities [128, 129] and specifically for data sharing within EHRs [94, 142]. Such regulatory complexity, in combination with the existence of different technical standards to address interoperability challenges [60, 93], has resulted in the development of many different nation-wide EHR architectures [36, 133, 136, 143]. On the other hand, each healthcare organisation, even when subject to the same regulatory policies and standards, can interpret and implement these policies and requirements differently in its internal IT environments. Therefore, for healthcare organisations to participate to cross-organisational EHR programs and to share medical data with others, they need to adapt their systems to satisfy a defined set of data management requirements and conform to the required EHR standards [9]. For these organisations, achieving interoperability is therefore a complex and challenging task. The challenges consist of:

- Analysing and mapping regulatory policies described in natural language into business-level specifications and then into privacy-aware enforceable data sharing mechanisms;
- Ensuring that data sharing mechanisms and privacy policies respect organisation specific requirements and best practices after the organisation joins the EHR sharing network;
- Developing, maintaining and evolving the IT infrastructure to support data sharing mechanisms and interoperability with other healthcare organisations.

We view the addressing of the these challenges and the development of compliance-aware data sharing mechanisms as a multidisciplinary task where business, IT and privacy experts need to collaborate towards the common goal of building the technology for sharing privacy sensitive data.

Contributions

To tackle these challenges, our approach is to develop a solution that can be offered as a service to healthcare organisations to facilitate the exchange of healthcare data. We want to equip organisations with the flexibility to accommodate a variety of regulatory compliance requirements and security and privacy policies, as these requirements and policies vary from country to country, from company to company, and over time.

Our key idea in addressing this lies in providing organisations with a methodology and the corresponding technology for mapping data owner's requirements into Business Processes Models (BPM) [139]. The resulting processes are then embedded into data management operations and executed inside the execution environment. As a consequence, organisations will be able to achieve compliant data management, and therefore cross-organisation and cross-regulation medical record sharing. More specifically, we propose:

- A **methodology** for establishing explicit links between high-level regulatory policies on one side, and detailed data management processes and privacy policies on the other;
- A **compliance-aware data management system** called CHINO that supports processes and policy definition and execution. As a result, CHINO manages (stores and shares) participant organisations' medical records according to their regulatory policies.

The methodology starts with the collection of business and compliance requirements that are later refined in the definition of executable data management processes and privacy policies. CHINO provides the technology to support the methodology and offers the modelling framework to enable participant organisations to model the business processes that implement the internal business logic of data management operations.

To evaluate the CHINO methodology and our system prototype, we examined regulatory and architectural differences among EHR systems and validated the prototype by integrating it with an existing medical record system called OpenMRS [146] and executing the identified policies. We tested process modelling usability with business process experts to ensure that they were able to easily define data sharing process models according to identified requirements.

Finally, with privacy experts we analysed how CHINO execution framework can achieve regulatory compliance in the Italian legislation context. Overall, the CHINO technology, and in particular its visual modelling representations of business processes, provides great visibility and transparency to security/compliance officers and business process analysts, and thus improves the trust, compliance and understandability among the participant healthcare organisations in terms of data sharing.

1.2.2 Architectures, Protocols and Policies for Privacy-Aware Sharing

While the previous research thread focuses on providing mechanisms to define regulatory compliant data management operations, this one is focused on the development of the underlying technology to support data management operations execution. Namely, this thread focuses on defining the operations interfaces, data sharing protocol, enforceable privacy policies, and components to store data and policies. The resulting contributions therefore, provide the necessary technology to support the process-based approach to defined data management operations described within the CHINO platform.

We start by analysing the challenges while enabling the cooperation among institutions delivering socio-healthcare assistance services. These services involve both healthcare and social assistance such as the delivery of assistance at home or residential care, and are provided by many public and private institutions. From a technical and organisational perspective, the development of data sharing solutions in this kind of scenario is challenging because:

1. The integration approach needs to depart from “traditional” data integration by requiring a light-weight process integration able to involve a large number of medium and small institutions that also dynamically grow over time (civic centres, hospitals and social care institutions will need to progressively join the initiative);

2. The exchanged information is privacy-sensitive containing patients’ social, health and financial data and making privacy policy definition challenging. Furthermore, the assistance processes are highly knowledge-based, sometimes not completely defined, spanning across several organisations and are executed by heterogeneous IT systems;

3. Strict privacy rules defined by data protection regulations and, most importantly, by guidelines for health records management [83, 94, 143] forbid the adoption of traditional data warehousing and integration approaches. Those rules and national-wide standards forbid collecting data in a central repository and define strict legal constraints on the way data is collected, stored and distributed in a context with multiple organisations.

Such constraints make it difficult to identify application protocols and policies for data integration. Since the Italian healthcare sector configuration and Data Protection laws are similar to many EU [69, 136] and non-EU [36, 76] countries, the problem is quite general.

To tackle the specificity of this integration scenario, the technology needs to preserve the organisation specific requirements and information systems, and at the same time, share data according to privacy laws and technical standards.

Contributions

We propose an integration approach that focuses on identifying and sharing *events* generated inside the assistance processes. The events contain the data generated by information system of participant organisations' within assistance process steps (i.e. while caregivers deliver some assistance). To reduce the disclosure of privacy sensitive data, we separate the events content into two kinds of data: *metadata* and *records*. While the records contain the detailed data content, metadata describe the records by reporting patient identity, when the record has been generated, and by whom. To share metadata and records, we provide an event-based protocol that delivers metadata to interested parties via a publish/subscribe mechanism [68], and then share the records only upon explicit requests which express the purpose of use. To limit data disclosure and deliver only the data that is necessary for the specified purpose of use, we give data consumers the option to specify fine-grained filtering policies over records, and to define which portions of records to be delivered to data consumers.

Overall, the proposed approach and technology provide the following contributions:

1. We show how a data integration problem in a multi-organisation and rapidly evolving environment can be addressed via an event-based approach. It makes it easy for new institutions to come on-board, and minimise the development and maintenance effort required for the integration;

2. We describe how privacy protection and data sharing can coexist by restricting the access to information only on demand and providing fine-grained privacy policies to constrain on who can see what and for which purpose;

3. We present the architecture and implementation of a solution that achieves integration of socio-health assistance services, and discusses the many lessons we have learned by deploying an instance of it in the Province of Trento, Italy, and testing it with the involved institutions.

The source code of the developed system has been also released under the GPL v3 free software license [77] and made available at the European Commissions Joinup repository [48].

1.2.3 Access Control Policy Violation Prevention

The previous research thread describes a data sharing protocol that relies on fine-grained access control policies to govern data disclosure. However, when data sources specify policies on their data (e.g. on their local schema or records), they cannot anticipate semantic inferences when data is integrated at the EHR level [131, 172]. Such inferences (e.g. given by functional dependencies which provide semantic constrains over attributes in a relational schema) can allow malicious users to obtain prohibited information by linking multiple queries and thus violate the data sources' policies. Therefore there is the need to proactively detect and prevent policy violations before the policies are deployed and enforced.

Contributions

We propose a set of algorithms that, given the EHR (relational) schema, the functional dependencies that holds on it, and the sources policies, are able to identify the sets of queries that, if linked based on the functional dependencies, lead to policy violations. These sets of queries that can lead to policy violations are also called violating transactions. To avoid the completion of a violating transaction, we propose a query cancellation algorithm that identifies a *minimum set* of queries that needs to be avoided. The identified sets of queries are then used to generate additional rules to be added to the existing set of rules provided by data sources to prevent the leakage of any prohibited information. We validated our algorithms on downloaded datasets from the web and synthetic datasets.

1.2.4 Summary of Contributions

In summary, the thesis reports three research threads that aim at proposing technologies for enabling privacy and compliance aware medical record sharing and thus, facilitate the creation of EHR systems. The Figure 1.1 depicts a high level EHR logical architecture and identifies the research contributions of each of the threads.

On top it shows the organisations interacting with the EHR to store and share data. To interact with the EHR, they need to develop adapters that support the interaction protocol proposed by the EHR (as detailed in Chapter 4). The adapters will receive and consume the data produced by other organisations.

Within the EHR architecture we identify three logical layers: data management interfaces (or Application Programming Interfaces - APIs), data management operations implementation (or business logic), and components that store data and policies and that can be internal or decentralised (e.g. external data stores).

The thesis contributions reported within the three research threads tackle different aspects of the overall EHR system architecture. Namely, the research thread 1 introduced in 1.2.1 and detailed in Chapter 3 focuses on the compliance within data management operations and it proposes a process-based approach and the related technology to define and execute operations internal business logic. We allow the organisations to define such business logic by the business process modelling framework.

The thread 2 in 1.2.2 and Chapter 4 focuses on developing the data sharing protocol, operations interfaces, the technology (i.e. the design of the internal components), and the policy

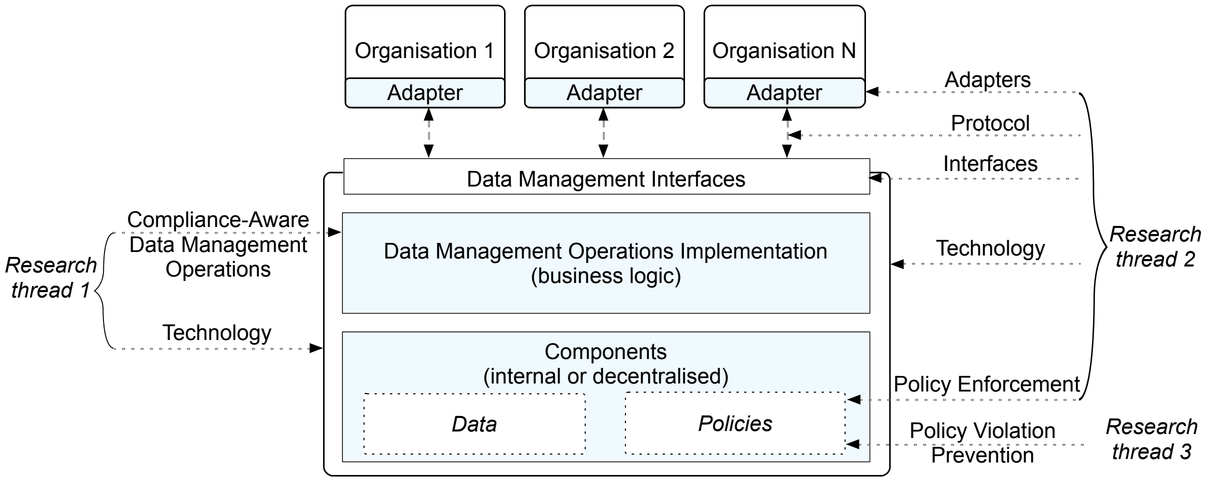

Organisation 1
Adapter
Organisation 2
Adapter
Organisation N
Adapter
Adapters
Protocol
Interfaces
Data Management Interfaces
Compliance-Aware Data Management Operations
Research thread 1
Technology
Data Management Operations Implementation (business logic)
Technology
Research thread 2
Components (internal or decentralised)
Data
Policies
Policy Enforcement
Policy Violation Prevention
Research thread 3

Figure 1.1: The presented research topics and their contributions in building EHR systems.

enforcement. In addition, as detailed in Chapter 4, it provides also adapters to organisations to facilitate the interaction with the EHR. Finally, the thread 3 in 1.2.3 focuses on challenges related to access control policy violation prevention.

1.3 Structure of the Thesis

The thesis structure follows the sequence of reported contributions and is organised as follows:

Chapter 2 gives an overview of research efforts in related areas. It starts by analysing some of important projects and initiatives in building EHR systems and technological standards. Then for each of the three research threads, it identifies the related work and state of the art solutions.

Chapter 3 describes our work done in the first research thread on compliance-aware medical record sharing. It starts by identifying regulatory differences among different states and proposed technologies. Then it shows how business process technology can be used to model data sharing processes and policies to achieve compliance. It reports the system validation phases by integrating with the existing EHR systems, performing a user study with BPM experts and by collaborating with privacy experts.

Chapter 4 reports our work in the second thread on EHR architectures and data sharing protocols to achieve privacy-aware data sharing. It describes the motivating scenario and challenges, the data sharing protocol, and the system prototype that has been developed and tested in Trentino, Italy. It also describes how the proposed technology provides the basis for the work described in Chapter 3.

Chapter 5 describes how we tackled challenges related to the access control policies violations in a data integration scenario. It shows how data inference (e.g. using functional dependencies) can disclose prohibited information, and how the proposed algorithms can help guard the design of the access control policies.

Chapter 6 summarises our research contributions and the lessons learned while developing data sharing systems in healthcare. It describes also some limitations and potential future extension on the identified three research threads.

1.3.1 List of Publications

The thesis contributions has been also published within the following peer-reviewed publications. The first three works describe the CHINO methodology and technology. The next three works (number 4, 5 and 6) report our contributions described in Chapter 4 while the work in 7 presents the research content of the Chapter 5. The remaining publications present some research efforts that are only partially related to the topics covered by this thesis.

1. J. Stevovic, J. Li, H. Motahari-Nezhad, F. Casati, and F. Armellin. Business process management enabled compliance-aware medical record sharing. *Int. Journal of Business Process Integration and Management*, 6(3):201 – 223, 2013. [164]

2. J. Stevovic, J. Li, H. Motahari-Nezhad, F. Casati, G. Armellin, and B. Farraj. Compliance aware cross-organization medical record sharing. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on,* pages 772 – 775, 2013. [165]

3. J. Stevovic, E. Bassi, A. Giori, F. Casati, and G. Armellin. Enabling privacy by design in medical records sharing. In *Pre-proceedings of In Reforming Data Protection: The Global Perspective*. Springer Netherlands, 2014. [163]

4. G. Armellin, D. Betti, F. Casati, A. Chiasera, G. Martínez, and J. Stevovic. Privacy preserving event driven integration for interoperating social and health systems. In *Proceedings of the 7th VLDB Conference on Secure Data Management, SDM'10*, pages 54–69, Berlin, Heidelberg, 2010. Springer-Verlag. [11]

5. G. Armellin, D. Betti, F. Casati, A. Chiasera, G. Martínez, J. Stevovic, and T. Toai. Event-driven privacy aware infrastructure for social and health systems interoperability: CSS platform. In *ICSOC*, pages 708–710, 2010. [12]

6. [1] A. Chiasera, J. Stevovic, F. Casati, G. Armellin, and D. Betti. Event-driven privacy-aware integration of social and health systems. In *Submitted to Journal of Medical Systems*, 2014.

[1] The work is not yet published and is currently under the review process.

7. M. Haddad, J. Stevovic, A. Chiasera, Y. Velegrakis, and M. Hacid. Access control for data integration in presence of data dependencies. In *S.S. Bhowmick et al., editor, DASFAA*, pages 203-217. Springer–Verlag, 2014. [87]

8. J. Stevovic, A. Maxhuni, I. Khaghanifar, G. Convertino, J. Shrager, and R. Gobbel. Adding individual patient case data to the melanoma targeted therapy advisor. In *PervasiveHealth*, pages 85–88, 2013. [166]

9. T. Peng, M. Ronchetti, J. Stevovic, A. Chiasera, and G. Armellin. Business process assignment and execution from cloud to mobile. In *International Workshop on Emerging Topics in Business Process Management*, pages 54–69, 2013. [150]

10. G. Armellin, D. Betti, S. Bussolon, A. Chiasera, M. Corradi, and J. Stevovic. From PHR to NHR? An UCD challenge. In *International workshop on Personal Health Record, Trento, Italy*, 2011. [10]

11. M. Corradi, A. Chiasera, G. Armellin, and J. Stevovic. Understanding how people work: experiences in improving healthcare practices in Italy. In *Workshop on Coordination, Collaboration and Ad-hoc Processes (COCOA'10)*, Palo Alto, CA, USA, 2010. [50]

Chapter 2

State of the Art

This section analyses the state of the art literature related to the topics and research areas covered by this thesis. Following the overall structure of the thesis, we start by analysing how the challenges related to regulatory compliance has been tackled within existing healthcare IT solutions. We focus in particular on the workflow based technologies and their adoption in healthcare to facilitate the regulatory compliance. Then we review technologies that have been proposed to solve interoperability and privacy related challenges in medical record sharing. Finally, we focus on access control aspects and challenges related to access control policy definition to protect patients' privacy.

2.1 Health Information Systems

Data sharing about EHR systems, and more in general, about Health Information Systems (HIS) represents one of the key features to enable such systems to exchange data and provide many benefits such as cost reduction, improvement of the quality of care, better control of adverse drug events and many more [89, 145]. To achieve data sharing, EHR systems need to provide mechanisms to interconnect different Electronic Medical Record (EMR) systems owned and managed by single points of care (e.g. OpenMRS [146]). Such mechanisms need to provide effective and efficient protocols and standards to collect privacy-sensitive data from single EMRs and implement data sharing mechanisms and protocols to deliver data to users (patients and clinicians) that need to access to them. The same data, after being properly anonymised, is used also for Business Intelligence over assistance processes to identify bottlenecks and monitor service delivery. The data can be used also for research purposes and for augmenting the knowledge about specific (and sometimes rare) diseases [166]. Another application of EHRs is to provide data for the creation of Personal (or Personally Controlled) Electronic Health Record systems (PHR) in which patients are able to create data (e.g. self monitoring) in addition to the possibility to visualise data produced by hospitals. By doing so PHR solutions, among the other benefits, are able to empower the patients and increase their knowledge about their disease management process [89]. An example of such systems is Indivo [113] project from Harward-MIT.

The importance of such systems has resulted in different projects an action plans by governments from all over the world. Namely, many nations have started national-wide projects aiming at financing and supporting EHR development [123]. Solutions have been developed aiming at providing common national-wide backbone infrastructure for interconnecting EMR systems used by single hospitals. The NHS - Spine in UK [133, 132], Canada Health Infoway [36], Dutch AORTA [136], Italian inFSE [143] are just some of examples of such initiatives and projects. One of such projects called CSS – Cartella Socio Sanitaria [11] developed to create health and social record in Trentino Province, Italy is part of this thesis. We describe the technology we developed and the research outcomes. The project source code has been released as open source code [48].

Other open source projects and initiatives such as Direct Project [169] or OpenEHR [170, 27] have been developed to harmonize the EHR standards at national levels and standardise the EHR content to be exchanged among organizations. Projects such as the EU cross-country epSOS [67] have been proposed to interconnect the Member States national systems and exchange patient summaries and ePrescriptions.

Developing such systems involves different types of challenges affecting the project development phases (from analysis do the development and runtime). Namely, starting from the early analysis there is the need for collecting the set of regulations and laws and technical standards that are related to the scenario that is considered by the project. This is challenging because regulations varies from states and countries and are defined at different levels. Once identified, the regulations and guidelines need to be analysed to extract design requirements and constrains. Later in Chapter 3 we report a subset of privacy policies that has been extracted from the HIPAA [76] and Italian Data Protection legislation while next subsection details on how regulatory compliance has been tackled by other works. Once analysed the regulations and identified the set of constrains and requirements, the system designers need to choose the appropriate technology that is able to enforce and satisfy them. Finally, the developed technology needs to be inspected by privacy and legal experts before going into production. This activity can lead to observations, and in worst case to rejection, of the developed technology. Those kind of inspections can be done also at runtime in case of legally motivated activities [84].

Before detailing on state of the art technologies and standards that has been proposed to build EHR systems, we will detail on how state of the art works have solved privacy-aware data sharing and, more in general regulatory compliance.

2.2 Regulatory Compliance

The impact of regulations on organizations' operating processes is one of the most important research topics nowadays. In healthcare, data protection laws, regulations and technical guidelines define different types of requirements on how data need to be collected, managed (stored and used for different purposes) and shared among healthcare providers or third parties. The intrinsic complexity of both regulations and healthcare sector makes the application and verification of regulatory policies an extremely challenging task.

Challenges start with the collection of the set of regulations and guidelines that need to be considered while developing an IT solution for a specific use case. For example, the Italian context is characterized by many levels of authorities which protect citizen's privacy rights: starting from the EU level legislations [69, 70] transposed in Italy with the Data Protection Code [83], to the Guidelines and recommendations provided by the Italian Data Protection Authority in collaboration with the Ministry of Health on Electronic Health Records [94, 143]. Then each region has its own competences on applying healthcare legislation, which is done by many local healthcare providers called "Azienda Sanitaria Locale (ASL)" that deliver assistance services to patients [128, 129]. This shows clearly that in Italy, but this applies also to other countries, there exist many bodies having different competences that define privacy legislations on different aspects.

Once identified the set or regulations, there is the need to "parse" them and extract policies and requirements. Different works have been done to in such direction to extract requirements in a systematic manner [9]. Usually those works describe approaches that can be applied to the translation of regulations into requirements or event further into design of compliant system functionalities according to rights and obligations expressed by regulations. Some of these works that we have identified include [33, 98, 159, 103]. The work [82] analyses regulatory aspects in a multi-jurisdictional and outsourced setting such as cloud-based solutions. Although the analysed scenario is similar to ours, enforcement aspects are not considered. The work [37] analyses privacy obligations and provides a framework for obligations specification and enforcement. Such approach is not sufficient to achieve the compliance-aware data sharing goal defined in this thesis, as it does not consider remote policy enforcement points or data stores. Other works provide an approach in designing legal patterns that can satisfy the legal requirements [101].

The extracted requirements define constrains on the HIS systems architecture and on their run-time behaviour [28]. For example, according to Italian Data Protection law [94], health care sensitive data should remain under the responsibility of the data controllers (i.e. entities that produce the data). Such constrains affect deeply HIS design choices as it is shown in our work [11]. The design phase therefore need to consider regulatory compliance which involves many challenges related to the definition, enforcement and monitoring of constraints extracted from regulations [11, 165, 164, 163]. These challenges are even more difficult to tackle in multi-organisation environments where multiple heterogeneous entities are involved [11, 164, 87, 140].

When HIS projects involve multiple institutions, one of the key factors for their success is given by identifying the appropriate project management methodology that is able to involve the stakeholders and domain experts during the elicitation and validation of the requirements [24]. Representing visually such requirements (e.g. by using diagrams) in a form that is understandable by the stakeholders, and useful to the developers to extract design requirements, can be of fundamental importance [29]. Languages and formalisms like workflow modelling, BPMN [139] and activity diagrams can be very powerful to model concisely complex interactions [62, 153]. We have applied successfully in one of our case studies described in Chapter 4 the extended activity diagrams. However, sometimes such representations are not sufficient and other frameworks could be needed to model the project scenario [13]. In one of our works in Chapter 3 we adopt BPM technology [139] to design and develop executable business processes and to face legal and organizational requirements [164], or to model and execute cross-organizational health processes [168].

Another important question is how to meet privacy regulations while managing health care data in outsourced environments. In such cases, a solution that manages health care data must meet business and compliance requirements. The business requirements provide usually the sequence of steps that need to be performed by the involved participants in a given scenario (e.g. the work in [141] analyses and represents the doctor-consultation use case with UML style state-diagrams). The compliance requirements instead, are usually extracted from regulations and sometimes are represented as simplified checklists [28], that has to be applied to the considered scenarios.

Our research focuses in particular on exploiting requirement-modelling languages and on designing execution environment to support compliant data management. In this direction, the GEODAC [109] framework provides both the modelling language and the run time execution environment to ensure data protection and apply the data assurance policies specified by the customers. The GEODAC work is focused on the modelling language for the service providers and the service customers to communicate their data assurance policies. It provides a policy specification language for the service provider and the service customer to communicate their data assurance policies but it does not provide policy enforcement points orchestration and processes visibility. The importance of process visibility aspect is emphasized by work [29], which demonstrates that with visual representation of business processes, systems could improve the trust, compliance and understandability of data management processes. To the best of our knowledge, nobody has applied the idea of modelling and defining the behaviour of compliance-aware data sharing operations by adopting process modelling languages as it is done in this work. In [164] we propose a model-driven methodology to collect requirements and define compliant processes and policies that are executed inside the proposed execution framework.

Another important approach that has been proposed to tackle the complexity of achieving privacy compliance is given by the **Privacy by Design** principles [40, 156, 39]. Privacy by Design principles has emerged recently as the most suitable approach in tackling privacy related issues and has been successfully applied in many projects and cases such as [23, 102]. Privacy by Design considers the privacy related aspects from early stages of systems design and has been introduced also by the Art. 29 Data Protection Working Party in the document The Future of Privacy [16] and the new European Data Protection Regulation [72].

Compliance-Aware Business Process Design is the approach we follow in our work described in Chapter 3. We start by observing that one of the main aspects that solutions that manage privacy-sensitive data need to ensure, is the transparency in data management [121]. Transparency is needed for regulatory reasons and to ease concern over the potential for data breaches. The work [46] states that one of the most important aspects of control is transparency in the implementation. Workflow transparency in one of the key contributions of this thesis since business processes technology is used to implement internal behaviour of basic data management functionalities [164] and 3. Some of the benefits of process visualization are better requirements satisfaction [29]. The authors demonstrate that visualizing operating processes can bring to better understanding and continuous improvement of regulatory compliance in the risk management field.

Another important aspect is given by the need of process validations through compliance checking and ensuring constrains satisfaction [154, 96]. The survey [96] gives a brief overview of the state of the art technology in business processes and compliance checking. Different approaches exist in enabling compliance within business processes. One approach is given by adding annotations to diagrams that are later transformed in executable mechanisms or languages. The work [52] provides a framework and a model-driven methodology for specifying security policies through process annotations. The final result is given by BPEL executable processes. The work[44] provides an integration of BPMN[139] with privacy policies to verify if the resulting process is compliant with P3P privacy policies. Other works such as[108] provide a language for expressing data retention policies that are required for compliance aware data management. It provides a language and a framework for their definition and execution. Those approaches give some advices and useful methodologies that will be further investigated.

Works such as [90, 118] annotates business processes with clauses, conditions and security policies that need to be satisfied at each process step. Instead we provide BPMN custom elements in addition to standard BPMN to model security, privacy and other constrains. Furthermore these works focus on verifying compliance on existing processes. In [26] process execution logs are analysed to evaluate privacy policies and disclosure of data in relation to process goals. Instead, this thesis focus is on process design to achieve compliance. The works reported by [110, 122] provide methodologies for inter-organization process design in the context of business contracts, while [21] synthetize process templates starting from compliance requirements. In contrast, our focus is on designing compliant processes that implement data management operations. However, these process design techniques can be certainly leveraged in our methodology to support developers and business analysts when developing regulatory compliant processes. The work [148] demonstrates how workflows can also improve compliance to clinical guidelines. A work that advocates the need of a different approach in using BPM technologies in healthcare is given by [115] where the lightweight processes, called proclets, exchange messages through communication channels and collectively implement healthcare processes. The work in Chapter 3 adopts a similar approach with a similar vision about the need of using lightweight processes to implement data management operations.

The compliance aspects can be ensured only if the underlying technology that provides data sharing protocols is safe from security and privacy points of view. In the next section we report state of the art technologies and protocols that has been proposed to manage healthcare data according to security and privacy constrains.

2.3 Architectures and Protocols for Medical Record Sharing

From a system design point of view, Service Oriented Architectures (SOA) [6] emerged as most commonly used paradigm for achieving interoperability in multi-organisation contexts. The SOA paradigm is adopted to tackle point-to-point synchronous interactions and it can become easily unmanageable in a scenario with many actors and systems impersonating dynamic roles and increasing in number. Event Driven Architectures - EDA [68] can solve such problem by decoupling service providers and consumers through asynchronous messaging. In our solutions described in Chapters 3 and 4 we adopted a mixed EDA-SOA driven approach [119] in which involved entities exchange data through WS invocation while the platforms implement the pub-/sub [68] functionalities through a Service Bus [126, 8] to exchange asynchronously data. A similar approach has been proposed in [125], as both approaches use events to transfer information by pub/sub although our case is more general because it is not limited to mobile devices. In Italy, a specific SOA based architecture called SPC [58] has been proposed to interconnect public administrations. We had as a requirement in our project the considering of such design guidelines and therefore the developed system described in Chapter 4 has been tested inside the SPC network. The SOA and EDA provide communication patterns and security standards to deal with data transmission.

Another important problem is how to represent, store and share privacy sensitive data. The Integrating the Healthcare Enterprise - IHE consortium [93] proposes a set of solutions targeting the semantic and syntactic integration aspects (applied to EHR projects such as Canada Health Infoway [36], Dutch AORTA [136], Italian inFSE [143], UK NHS [133], European cross-country epSOS project [67]) based on a central registry of searchable meta-data linked to the data generated by producers (typically encoded as HL7 and CDA [60]). Various implementations of registries exist such as UDDI, ebXML, XML/EDI [180]. The most appropriate in terms of flexibility, interoperability and adoption for the healthcare domain [64, 59] is ebXML by OASIS [124] that we adopted in our project by using the freebXML instance [78]. Privacy regulations impose constrains (not always well defined) on the design of the health information systems and on their run-time behaviour. For example, according to [94, 143] healthcare sensitive data should remain under the responsibility of the data controllers. In such systems, the definition, enforcement and monitoring of privacy constraints [43] is fundamental especially in multi-domain environments where multiple heterogeneous entities are involved [140]

At the interoperability layer, many types of solutions have been proposed to manage healthcare data. In particular recently we observed a high adoption of BPM technologies with the aim at improving interoperability and service quality [105]. While works such as [42] proposes BPM to model cross-organization or cross-department processes, we propose the use of BPM to model and execute internal operations over data. As shown in Chapter 3, BPM in addition to compliance, can facilitate also interoperability and building more easily the technology for medical record management in cross-regulation settings.

The technology is necessary to ensure secure and safe medical record management. It ensure the delivery of information according to rules that define the users' access capabilities and rights on medical records. Therefore, to achieve privacy-aware data management, access control is fundamental. Next subsection analyses state of the art solutions to implement access control in healthcare settings.

2.4 Access Control for Medical Record Sharing

There exist several approaches to tackle the challenges of securing the access to data in multi-organisation environments. **Privacy preserving disclosure** strategies assumes that as long as the data has not been published yet and is stored on a trusted server, which thus is not vulnerable to attacks[79], the data is safe. After publishing, the data cannot be protected anymore, and therefore it need to be made less privacy-sensitive. Works in such direction such as *k*-anonimity, *l*-diversity, anonimity-plausibility, limited retention, data degradation and many more are accurately analysed by this survey[79].

The work done in this thesis (in all the three chapters), approaches the privacy related challenges by ensuring appropriate access control. Access control is part of the **disclosure prevention** strategies that have been proposed by server-side mechanisms such as access control, P3P, and server-side encryption or applied by client-side mechanisms such as P4P [61] and client-side encryption [75]. Disclosure prevention provides security mechanisms mainly through access control and encryption to prevent unauthorised access. This aspect is currently becoming of high importance in cloud based environments, where the main challenge consist of protecting data in untrusted environments [38]. A common approach is to outsource entirely encrypted data that prevents unauthorized intrusions. This approach forbids the performing of searching operations over data. Some of the revolutionary research works provide mechanism to perform search operations over encrypted data with fully homomorphic encryption schemas[80]. Another issue is how to manage encrypted data and the related encryption and decryption keys at large-scale. A mechanism and full implementation of an efficient and scalable data- and key-store is proposed by [108]. In our works we focus on defining access control policies such that only authorized users can access to certain medical information.

Purpose Based Access Control provides an efficient privacy disclosure prevention methodology to achieve a balance between privacy and utility of data. In fact, in healthcare an important aspect that needs to be satisfied is the **contextual integrity** which measure the closeness of the conformity of personal information with context-relative informational norms [25]. For example, medical information shared with someone outside the health-care context represent a violation of privacy. The work [26] provides a methodology for designing business processes and privacy rules that satisfy privacy goals and organizations utility. This approach can be useful to complete our model-driven approach. The work [95] describes a interesting approach for delivering data on a purpose based need-to-know basis. This is achieved through the definition of purposes of use. With [11] we propose access control policies to preserve privacy in data sharing in an event-driven environments. We define in particular fine-grained access control policies based on a purpose of use of data. The purpose-based access control is normally considered a good solution for meeting the requirements of privacy legislations. The purpose taxonomy for the healthcare domain is well defined at national level in Italy [94]. It has been preferred to identity-based access methods, such as RBAC, because of its efficiency and suitability in multi-organization and variable context that usually present the role explosion problem. For that purpose, RBAC extensions such as P-RBAC[135] has been proposed that extends RBAC with privacy annotations such as purposes, event-condition-actions and obligations.

In Chapters 3 and 4 we defined a protocol and privacy policies for data sharing in event-driven [68] environments that satisfy the data sources needs. In [164] these mechanisms are incorporated into the business process execution in order to achieve privacy-aware data sharing. In [11] we apply a purpose-based access control mechanism that has been favoured by the identity-based access control frameworks such as Role-Based Access Control (RBAC) [155], because of its suitability in multi-organization and variable contexts [135]. In fact, RBAC policies such as Ponder [54] present the role explosion problem that does not apply to purpose-based mechanisms such as Privacy-Aware Role-Based Access Control (P-RBAC) [135]. In P-RBAC and similar works, an intended purpose is defined for each data object specifying the intended usage of that data object. The access purpose is attached to each request and only when an access purpose is compliant to its intended purpose the access is allowed. This approach has been adopted also in [11] with some differences such as definition of sub-document level policies.

In other mechanisms such as the one presented in [95], the purpose is used to define patient-centric authorization model and categorization and policy definitions. Such mechanisms could be considered to further improve CHINO data disclosure. In systems such as CHINO many challenge are related with protecting sensitive data in untrusted environments [46]. Security in exchanging data is approached by many works such as [181] that proposes a system to securely exchange healthcare data in peer-to-peer fashion. The proposed system is based on cryptography and secure key sharing. Our work instead proposes centralized policy management and aims at enforcing also organization specific requirements. Our previous work [108] considers the key management issues and provides a mechanism to support an efficient and scalable encryption key store that we will consider for future CHINO extension.

An important challenge is given by how to represent access control policies in a SOA/EDA environment. A variety of **policy representation standards** have been proposed by using XML language [116] including the definition of purpose based policies. Policy specification languages such as P3P [182] and access control languages such as IBM's EPAL or, previously mentioned, OASIS' XACML [124] allow users to express privacy requirements in terms of the authorized purposes of the data when it is released to a third party [7]. A standard that better suits in distributed and inter-applications communication scenario is given by eXtensible Access Control Markup Language - XACML [124]. Pure XACML is used as the policy specification language for both document-level and attributed-level access control mechanisms [183]. To best fit to web services it has been extended to support a fine-grained security enforcement [45]. Some approaches rewrite queries [111] while the work [53] proposes a fine-grained solution for the definition and enforcement of access restrictions directly on the structure and content of the documents providing a specific XML response with its Document Type Definition - DTD. In [11] we apply the same idea using XML schema - XSD instead of DTD as it is more suitable for Web Service invocations. There are also other type of approaches in specifying access control based on the specified purpose such as [95] which defines an interesting approach in categorization and policy definitions. Its suitability for this research work will be further investigated. To enforce XACML policies we use an engine written in Java language [66]. Although these access control mechanisms are sufficient for healthcare scenarios to exchange data among authorized parties, they may not be suitable to other scenarios where data inferences could be used by malicious users to mine prohibited information.

Access control in information integration scenarios represents a challenging and fundamental task to enforce the appropriate authorization policies [30, 134]. In particular in federated database scenarios where sources are exposed through a mediator as one single database (see [158] for a survey). In data integration systems [106], a set of sources are exposed through a mediator as one single database. The mediator offers a unique entry point to all sources. In such system defining access control policies is a challenging and fundamental task to be achieved [30]. In work shown in Chapter 5, to define access control rules we rely on authorization views that are expressed by means of datalog rules. In [5] the authors consider the use of metadata to model both purposes of access and user preferences. We do not consider these concepts explicitly, but they could be simulated by using predicates while defining the access control rules. The authors of [57] analysed different aspects related to access control in federated contexts [158]. They identified the role of administrators at mediator and local levels and proposed an access control model that accommodates both mediator and source policies. In [104], the authors propose an access control model based on both allow and deny rules and algorithms that are able to check if a query containing joins can be authorized. This work, like the previous one, does not consider any association/correlation between attributes or objects that can arise at global level when joining different independent sources.

Sensitive associations happen when some attributes, when put together, lead to disclosure of prohibited information. Preventing the access to sensitive associations becomes crucial (see, e.g., [4, 47]) in a distributed environment where each source could provide one part of it. In [4], the authors proposed a distributed architecture to ensure no association between attributes could be performed while in [47] fragmentation is used to ensure that each part of sensitive information is stored in a different fragment. In [55], the authors propose an approach to evaluate whether a query is allowed against all the authorization rules. It targets query evaluation phase while our goal is to *derive additional authorization* rules to be added to the mediator.

Inferences induced by data dependencies such as functional dependencies (FDs) are difficult to identify and solve while defining or applying privacy policies (see [74] for a survey). In [179], the authors provide an anonymisation algorithm that considers FDs while identifying which portion of data needs to be anonymised. In our case, we focus on defining access control policies that should be used to avoid privacy breaches instead of applying privacy-preserving techniques [79]. In [130], the authors proposed an approach to automatically generate, from access control policies defined over database relations, the access control policies that are needed to control materialized views to ensure data confidentiality. Although this work allows for inferring policies it does not consider functional dependencies.

In [56] and [167] for each inference channel that is detected, either the schema of the database is modified or security level is increased. In [56], a conceptual graph based approach has been used to capture semantic relationships between entities. The authors show that this kind of relationships could lead to inference violations. In [167], the authors consider inference problem using FDs. This work does not consider authorization rules dealing with implicit association of attributes; instead, the authors assume that the user knows the mapping between the attributes of any FD. Other approaches such as [35, 171] analyse queries at runtime and if a query creates an inference channel then it is rejected. In [171], both queries and authorization rules are specified using first order logic. While the inference engine considers the past queries, the functional dependencies are not taken into account. In [35], a history-based approach has been considered for the inference problem. The authors have considered two settings: the first one is related to the particular instance of the database. The second is only related to the schema of both relations and queries. In our work, we focus on inferences to identify additional access rules to be added to the mediator. In [85], we investigated how join queries could lead to authorization violations. In this thesis, we generalize the approach in [85] by considering the data inference problem.

Another related challenge is given by the background, external or adversial knowledge which refers to the additional knowledge an user may have while querying a source of information. Works such as [117] and [41] analyse aspects related with combining the retrieved knowledge from a system (e.g. Mediator) with external knowledge. These aspects, similarly to inferences, are very important and need to be considered during the definition of privacy policies to avoid cases such as Netflix [131] where privacy sensitive information has been inferred from anonymised data.

The challenges related to **discovering functional dependencies** in database systems has been considered by many works due to their importance during design phase [114]. In our work we focus in particular on functional dependencies that do not hold on all the tuple of a relation. This kind of functional dependencies has received different names. In [178], probabilistic functional dependency are used to normalize a data integration schema constructed automatically. In [91], the author discuss discovery of both functional dependency and partial function dependencies. Their approach is based on the definition of equivalence classes within attributes values to check the validity of a functional dependency or partial one. Then, the classical lattice based method is used to avoid checking all the possible combination of attributes.

Another problem related to functional dependency is how to infer a set a functional dependency on a view. This problem has been studied in [99] and [100]. More recently [73], the same problem has been studied for a specific class of functional dependency. Namely, condition function dependency. In [97], different measures of partial functional dependencies are discussed. In our work we consider the $g3$ measure. These works has been considered and could be used in future to extract the exact sets of FDs from involved sources schemas.

Chapter 3

Compliance-Aware Medical Record Sharing

This chapter presents our work on tackling the challenges related to medical records sharing in cross-organisation and cross-regulation settings.

It starts by analysing differences between the Italian and UK regulatory contexts and organisations' specific requirements. Then it presents the CHINO Methodology and execution framework, and describes how organisations can define compliant data sharing processes and policies. It shows also how processes and policies are executed within the CHINO execution framework.

Before concluding, it presents the following validation phases: (i) the CHINO integration with OpenMRS to demonstrate technical interoperability, (ii) the usability study with business process experts to demonstrates that process modelling is a feasible task within CHINO, and (iii) the review of CHINO features and capabilities with respect to Italian privacy regulations by collaborating with a privacy expert.

3.1 Introduction

To achieve medical record sharing, organizations need to deal with regulations that define rules on healthcare data management at different levels, starting from unions/federations on to individual countries and regions. Such regulatory differences have resulted in the definition of various interoperability standards and nation-wide Electronic Health Records (EHR) architectures to interconnect healthcare organizations [69, 132].

In addition to privacy concerns, organizations also have their own business requirements and needs in terms of healthcare data management, which result in custom data representations and specific security protection mechanisms. Therefore, in order for healthcare organizations to share medical data with others, they need to adapt their systems to satisfy a defined set of data management requirements in order to conform to the required EHR compliance policies. For these organizations, achieving interoperability and exchanging data in compliance with regulatory policies is therefore a complex and challenging task [9]. In particular, the challenges consist of:

- Mapping regulatory policies described in natural language into business-level specifications and then into privacy-aware enforceable data sharing mechanisms;
- Ensuring that data sharing mechanisms and privacy policies respect organization specific requirements after the organization joins the EHR sharing network;
- Developing, maintaining and evolving the IT infrastructure to support data sharing mechanisms and interoperability with other healthcare organizations.

The goal of our work is to design a common data sharing solution that can be offered as a service to healthcare organizations to facilitate the exchange of healthcare data in a compliance-aware manner. Existing commercial EHRs such as Practice Fusion [151], Microsoft HealthVault [120] and other emerging cloud-based solutions today do offer medical record management, but only according to a fixed set of regulatory policies and technical standards. Instead, we want to equip organizations with the flexibility to accommodate a variety of regulatory compliance requirements and security and privacy policies, as these varies from country to country, from company to company, and over time.

However, flexibility often comes with increased complexity. Namely, the key challenge in providing greater flexibility sits in not increasing the difficulty of the system usage.

Our key idea in addressing this trade-off lies in providing organizations with a methodology and a platform based on mapping data owner's requirements into business processes which are then embedded into data management operations. As a consequence, they will be able to achieve compliant data management and cross-organization and cross-regulation medical record sharing.

More in detail, we have i) identified a methodology for establishing explicit links between high-level regulatory policies on one side and detailed data management processes and privacy policies on the other, and ii) designed and developed a compliance-aware data management system called CHINO that supports processes and policy execution.

The methodology starts with the collection of business and compliance requirements that are later used in the definition of executable data management processes and privacy policies. CHINO provides the technology to support the methodology execution and offers the following unique features:

1. It provides the components and the modelling framework to enable participant organizations to define the business processes that implement the internal business logic of data

management operations. The processes manage medical records according to organizations' regulatory compliance requirements and security and privacy policies.

2. To facilitate the business process modelling activity and the policy enforcement, it provides i) a set of custom elements that facilitate access to low-level operations on data and rules managed by the internal IT infrastructure components such as record store and metadata registry and ii) a set of policy enforcement templates that can be combined to model the data sharing processes while enforcing identified policies.

3. To execute the organizations' processes and policies it provides a shared execution environment that supports data sharing across participant organizations.

To evaluate the CHINO approach and prototype, we examined regulatory and architectural differences among EHR systems and validated the prototype by integrating it with existing EHR systems and executing the identified policies. In particular, we analysed some common cross-organizational data sharing scenarios in Italy and UK. We defined data sharing processes in compliance to Italian and UK regulations and executed them inside CHINO. To test the data sharing scenarios, we integrated CHINO with an open source medical record system called OpenMRS [146]. In the integrated system, the data sharing processes are used to mediate two OpenMRS instances belonging to the two regulatory contexts and having their own data management processes and policies. This integrated system demonstrates that with CHINO, organizations are able to share medical records while being compliant with regulations and satisfying their internal business requirements.

We report also the usability study in which we have validated the CHINO Modelling framework usability with nine business process exerts to verify if they were able to model data management processes and policies according to identified set of requirements.

Our solution takes advantage of the techniques that have been developed for business process modelling and execution, and applies them into the compliance-aware data management domain. In particular we propose processes to implement granular data management operations logic.

With the collaboration of a privacy expert, we reason about how business processes, and in particular their visual modelling representations, can provide great visibility and transparency to security/compliance officers and business process analysts, and thus improves the trust, compliance and understandability among the participant healthcare organizations in terms of data sharing [29]. Namely, we view the development of compliance-aware data sharing mechanisms as a multidisciplinary task where business, IT and privacy experts need to work towards the common goal of defining methods for sharing privacy sensitive data.

The focus of this and the following chapters is on the mechanisms and solutions to ensure privacy and other compliance requirements and not the actual content exchanged. When used to manage qualified content from the semantic point of view, our service is able to achieve full interoperability [149].

The rest of the chapter is organized as follows. Section 3.2 analyses regulatory compliance issues and provides examples to show how data sharing regulatory policies vary among different regimes. Section 3.3 presents the CHINO methodology to address compliance issues and define data management processes. The elements that characterize the CHINO modelling framework are detailed in Section 3.4. Their usage in enforcing the identified policies is described in Section 3.5 through the definition of policy enforcement templates. Process examples describing the templates usage are shown in Section 3.6 while Section 3.7 describes the CHINO architecture and prototype implementation details. The validations are discussed in Section 3.8 where we show the integration between CHINO and OpenMRS in subsection 3.8.1. We describe the User Study with whom we validated the Modeling Framework usability in subsection 3.8.2 while in subsection 3.8.3 we discuss how CHINO can achieve regulatory compliance. We conclude this chapter in Section 3.9.

3.2 Motivating Scenario

To understand requirements and policy differences among regulations and EHR standards, we analyzed some common cross-organization data sharing scenarios in Italy and UK. Such scenarios in practice are very important due to the current EU plans that aim at offering to citizens integrated services across EU countries with projects such as epSOS [67]. Cross-organizational data sharing can also cover the situation of cross-country data sharing.

The scenario we describe here is about doctor consultation. It starts with the patient requesting a visit to her personal doctor regarding a diabetes problem. The personal doctor then requests a consultation from a diabetes expert (specialist). Once the specialist visited the patient and gave his therapy suggestion, the doctor can prescribe appropriate medication to the patient. While doctors and specialists may belong to different healthcare organizations, both need to access the patients' medical records and exchange data. To conform to privacy policies they need to access only the minimum set of information necessary to carry out their tasks.

Our analysis of such scenarios focuses on the operating processes that cover policies and interaction protocols between these actors. We paid particular attention on differences with respect to privacy policies and entities responsible for applying them. The patients (or data subjects) usually agree on participating to EHR programs and accept the terms the data is accessed and shared. The EHR systems (or data controllers) produce and manage patients' records. The entity that is responsible for authorizing the disclosure of data (that we call data owner) varies according to regulations, the context and motivation with which the data is accessed. The owner can be either the data subject, controller or a third party in the cases when for example the pa-

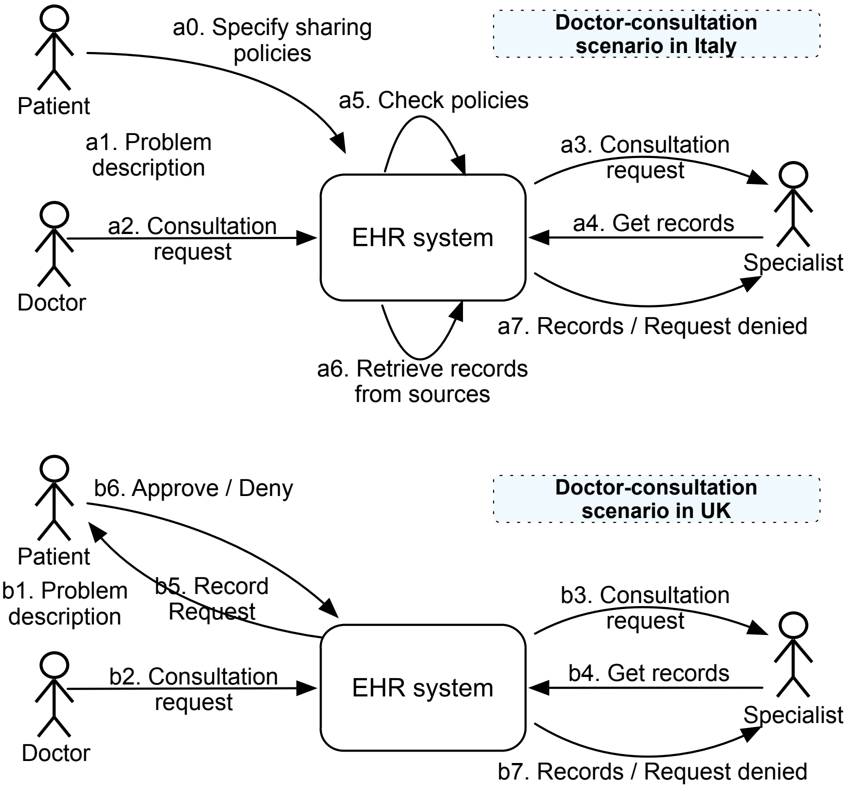


Figure 3.1: The doctor-consultation scenario as it is performed in two different regulatory contexts and EHR systems.

tients are under certain age or the records are about mental health [69].

In Italy, the Italian Ministry of Health defined a national wide architecture for interoperable EHR systems [143] along with data exchange standards and protocols to meet EU and national regulations on personal data protection [69, 83]. The hospitals act as data owners that are authorized by data subjects (the patients) to collect and disclose their health records to other healthcare organizations when needed. The patients can agree on data access policies for specific categories of healthcare operators and to grant access rights to records [69, 83]. To support information discovery, a hierarchical structure of metadata registries provides searching functionalities over decentralized record stores administered by individual hospitals.

In contrast, in UK the NHS - Spine [132] system is based on a centralized architecture for data discovery and storage. Patients explicitly grant access rights to requesters at each access to their medical records. In both contexts under certain exceptional situations, such as emergency or legally motivated cases, their authorizations can be overwritten and data is disclosed automatically [142].

Figure 3.1 shows a high-level interaction diagram that summarizes how information sharing and interactions are carried out among actors according to compliance policies and EHR systems in Italy [143] while Figure 3.1 shows how the same use case scenario is performed in UK [132]. Beyond the observable differences with respect to the interaction patterns, other key differences include:

1. **Security policies**, which define access control mechanisms, data encryption strategies and locations where the records should reside. In Italy, the records are stored in decentralized record stores inside individual hospital data centres while in UK they are stored at a backbone centralized record store. This implies the need for different data retrieval processes (a6) to retrieve the record from the record stores.

2. **Privacy policies**, which define who can access which data under which conditions and for which purpose. Conditions and purposes are usually defined by regulations along with exceptional cases [83, 142]. Regulations identify also who need to act as data owner applying data access policies and acting as Policy Enforcement Point. In Italy, the policies are defined at the record creation phase (i.e. when patients accept to create their EHRs at step a0 in Figure 3.1) and are applied by the EHR system on data requests (a5). Therefore the EHR system acts as the enforcement point and provides mechanisms to take decisions (a5) on requests [143]. In UK data owners apply the policies at runtime (b6) deciding to allow or deny the access to their data [142]. These policy enforcement differences result in different interaction protocols among different actors and systems.

3. **Business specific requirements** can be any organization-specific privacy, security, or other technological related requirement. Such requirements can represent obstacles for organizations in participating to EHR programs [159]. For example, organizations are required to adopt specific audit strategies. Therefore the EHR should be able to interact with the internal organization systems to send audit information in an appropriate format. Furthermore, the data retrieval process (a6) could consist of the invocation of organization specific Web Services following different standards (e.g., SOAP, REST and JMS).

Similar differences can be identified in other data exchange scenarios such as emergency room case or legally motivated cases in which public authority requests can override owners' policies. In such exceptional cases specific audit strategies need to be applied and records are disclosed to requester without restrictions [83, 142]. Consequently, the special conditions and auditing strategies have to be implemented and made transparent to auditors and privacy experts. These exceptional cases lead to regime-specific solutions that are not cost-effective, since the participating organizations need to adapt their systems and they become hard to maintain as rule changes.

Table 3.1 summarizes the policies that have been identified during the regulatory analysis and that will be later used to show how the proposed CHINO framework can express and enforce

Category	**Policy**	**Description**
Security	*Identificationi*	All entities that access to data must be identified.
	Safeguard	Collected data should be kept secure from any potential abuses.
	Access	Data subjects should be allowed to access their data and make corrections to any inaccurate data.
	Accountability	Data subjects should have a method available to them to hold data requesters accountable.
	Integrity	Data integrity should be verified upon retrieval and transmission.
Privacy	*Consent*	Data can be disclosed to third party when an entity (subject, controller or somebody else) responsible for data management, called owner, gives explicit consent. In special cases data can be accessed without the owner consent (e.g. compliance with legal obligations, protect vital interest of data subject, public interests).
	Purpose	Personal data can only be processed for specified explicit and legitimate purposes and may not be processed further in a way incompatible with those purposes.
	Proportionality	Personal data may be processed only insofar as it is adequate, relevant and not excessive in relation to the purposes for which they are collected and/or further processed.
	PIStorage	The data shouldn't be kept in a form that permits identification of data subjects.
Business Specific Requirements	CustomStandard	While satisfying all the previous requirements, data controllers can choose data representation standards and implement data management solutions according to their needs.
	PersonalAudit	Data controllers are responsible for establishing auditing strategies to ensure accountability on all operations over data.
	PrivateStore	Data controllers can be obliged to store records they produce inside specific administrative locations.

Table 3.1: List of identified policies and organization specific requirements and their description.

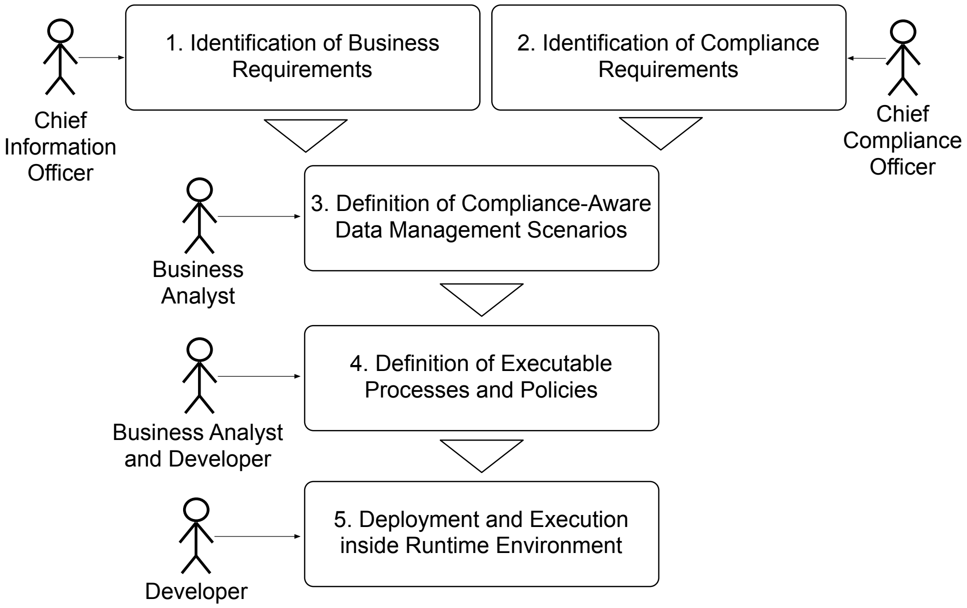

1. Identification of Business Requirements
2. Identification of Compliance Requirements
Chief Information Officer
Chief Compliance Officer
3. Definition of Compliance-Aware Data Management Scenarios
Business Analyst
4. Definition of Executable Processes and Policies
Business Analyst and Developer
5. Deployment and Execution inside Runtime Environment
Developer

Figure 3.2: The CHINO methodology and approach to define, translate, deploy and execute the data management operations.

them. The list is not meant to be exhaustive although it covers the most important policies identified in both regulatory contexts that we have analysed.

3.3 CHINO Methodology

The data sharing scenarios and policies presented in Section 3.2 demonstrate that a common EHR system involving different healthcare organizations needs to support different regulatory policies, interaction protocols, as well as organization specific requirements. We cope with the problems identified above by defining a methodology to support compliance-aware data sharing processes with organization-level customizations to manage (store, retrieve, and share) the data.

The methodology, as shown in Figure 3.2, consists of a sequence of steps performed by different actors:

1. First, the Chief Information Officer of the organization identifies the business requirements describing, for example, the flow of interactions, business specific requirements, and assigning flow steps to be fulfilled by different departments or organizations. Such requirements are often described in natural language with operational models describing how actors interact among them and with the EHR systems [132, 143].

2. Second, the Chief Compliance Officer of the organization reviews the business require-

ments, and follows the compliance checklists to identify the necessary compliance requirements, and security and privacy policies that need to be incorporated. For example, to define at which step which security and privacy policies need to be applied, and to identify exceptional cases in which data can be disclosed without patients' authorizations [83]. Examples of such checklists extracted from UK regulations can be found at Appendix B of [142].

3. Third, a Business Analyst combines the business requirements and the compliance requirements to devise a high-level representation that describes the steps that the involved parties should follow [159]. Figure 3.1 reveals examples of such representations (with many details omitted). The business analyst can also annotate the interaction diagram with the corresponding security and privacy policies identified at Step 2 [105].

4. Fourth, the Business Analyst and System Developers translate the high-level representations into executable business processes and rules. The business processes implement the business logic of granular data management operations such as PushRecord, GetRecord and so on. The operations reflect the identified compliance-aware data exchange interaction requirements and policies. Policies are then translated into security and privacy rules that are incorporated into the business process steps and enforced through operations on internal CHINO components.

5. Finally, the resulting executable business processes and rules are deployed and executed into the shared execution environment.

The processes orchestrate multi-party human and system interactions including patients, doctors and EHR systems. Business process steps access data through a set of operations that are executed on internal IT system components such as medical record store and metadata registry. Business process steps perform also the operations in terms of enforcing the defined security and privacy rules at policy enforcement points. Some components such as data stores or policy enforcement points can be also remote.

In summary, the CHINO methodology identifies the sequence of steps carried out by multiple stakeholders, from high-level business requirement collection to the low-level process execution and policy enforcement. This work focuses on Steps 4 and 5 of the methodology, i.e., on providing the technology to support the modelling and runtime aspects. The CHINO technology, together with the methodology, proposes a way of tackling compliance issues in multi-jurisdiction data-sharing scenario.

The analysis of activities related to requirement collection is discussed in several state-of-the-art approaches such as [29, 122, 143, 159] and is not discussed further here. This work aims instead at providing tools to enable involved actors, each one with its own skills, to collaborate with others in defining compliant data sharing processes.

Collectively, the business processes execution is able to meet the high-level compliance requirements. The steps can be performed at design and/or runtime to define and improve the compliant processes and policies. The methodology does not aim at identifying any specific software development model (waterfall, spiral or agile), yet it is intended to encourage process and policy improvements over time. In particular, the visual representation of policies should facilitate process tuning over time when policies change.

Although similar methodologies have been applied to solve compliance issues in business contracts and finance reporting [122], to the best of our knowledge, none of them solved the regulatory compliance issues in the domain of EHR related data management. In particular, none of them approached the implementation of data management operations through the definition of business processes, as it is the case in this work. In the rest of this chapter, we will present a modelling framework to capture and execute the compliance requirements identified during the first three steps of our methodology.

3.4 CHINO Policy Modelling Elements

Following the methodology, once the business analysts define high-level compliance requirements (Step 3), the analysts and developers translate them into executable business processes and privacy and security rules (Step 4). Overall, we adopt an artifact-centric process modelling approach [137], where the definition of the process models has the objective to access and share medical records. Such management has to be compliant with the policies identified at previous steps. To support this objective, the CHINO platform offers a set of technical features and components that are grouped in the following set of elements: *data, rules, Data Management Interfaces (DMI), modelling elements and Low Level Operations (LLO).*

The following subsections will detail each of these elements while Sections 5 and 6 will show how their combination can achieve compliance-aware data management according to the data management scenarios identified previously at *Step 3*.

3.4.1 Data

The establishment of a common understanding of terminologies of different policy administrative domains is key to interoperable data sharing [11, 93]. Our work is focused on the development of a framework for the definition of compliant data management operations and corresponding data sharing processes. We avoid forcing the use of particular message content representation standard. Although HL7 [60] represents the de-facto standard for encoding and exchanging healthcare data, other types of content (e.g. administrative and financial) need to be encoded in different formats. An example of such scenario is shown in the Case Study 4.2 in which organizations from social and health domain needed to exchange data.

```
<Record>
      <recordId>41</recordId>
      <recordTimestamp>2012-07-12 T 15:00</recordTimestamp>
      <recordType>22</recordType>
      <recordDesc>ConsultationRequest</recordDesc>
      <content>
            <crRequesterId>1</crRequesterId>
            <crRequesterName>John Watson</crRequesterName>
            <crConsultantId>2</crConsultantId>
            <crConsultantName>Jack Sheppard <crConsultantName>
            <probDesc>Lorem ipsum dolor sit,
            consectetut..</probDesc>
            <pi>
                  <patientId>2900EG-8</patientId>
                  <name>Rose</name>
                  <surname>Ledama</surname>
                  <gender>F</gender>
                  <birthDate>1980-05-15</birthDate>
            </pi>
      </content>
</Record>
```

Figure 3.3: An instance of the ConsultationRequest XML record.

With CHINO we provide a platform to share any data format and we leave to the organizations to rely on the state of the art in the literature to address semantic interoperability at the information model level.

To exchange data we propose a protocol based on IHE - Cross-Enterprise Document Sharing (XDS) profile concepts [93]. XDS has been proved as effective to exchange medical records in many projects [36, 143] and it has been adopted in our previous work in building an EHR architecture in the Trentino region - Italy [11]. The protocol is described with more details in Chapter 4 and in particular in Section 4.3.

To provide data discovery functionalities and to limit data sharing only to the data that is relevant to the participant organizations, XDS introduces the distinction between two kind of data that are called respectively *metadata* and *records*:

- *Metadata (M)*: describes a record and is used to build the patient medical history. It contains only the data necessary to identify a person (who), a description of what occurred (what), the date and time of occurrence (when) and the source of the event (where). *M* is necessary to discover information about a patient and is stored centrally on the metadata registry.

- *Record (R)*: contains detailed and privacy sensitive information (i.e. the consultation request shown in Figure 3.3). Records can contain any type of content (healthcare, administrative, financial) and can be stored centrally on the CHINO record store or kept on external data stores. To ensure policies such as Safeguard and PIStorage, records that are stored on the CHINO record store are encrypted. Although CHINO provides a central record store, its main role can be seen as a broker for data management aspects. Due to the existence of policies such as PrivateStore, it does not impose a centralized EHR management system. However, as the current trends are demonstrating, centralized cloud-based storage environments represent already one of the main cost saving strategies in healthcare [184].

The data exchange protocol proposed by XDS shares first metadata and then, only upon explicit requests, it shares records with interested data consumers. This approach limits the disclosure of privacy sensitive information and the amount of transferred data since metadata contains only a brief description of the corresponding record (see Section 4.3 for more details). To manage authorizations and access rights to records and metadata, CHINO provides a set of rules that data owners need to define for interested data consumers.

3.4.2 Rules

Rules are designed to implement a privacy preserving access control mechanism over metadata and records. An appropriate definition of rules is key to enforcement of the policies related to the access to data such as *Consent*, *Purpose* and Proportionality. To enforce all identified policies, we identified and designed two types of rules: access rights and data filtering rules. The *access rights (Ar)* rules define who can access a record or metadata while the *data filtering (Fr)* rules are used to disclose only the information that is required to perform a specific task. In particular, *Fr* is necessary to enforce the *Proportionality* policy, as we will show later. Overall, the data sharing protocol uses *Ar* to check authorizations and *Fr* to ensure privacy-aware data disclosure and in conjunction, these two types of rules implement a purpose-based access control mechanism. *Access Rights Rules (Ar)* ensure access control over metadata *M* and records *R*. The design of *Ar* has been inspired by the Privacy-Aware Role-Based Access Control model [135] that has been proposed as an extension of classical Role-Based Access Control models introducing the concept of purpose of use of data. CHINO assigns to both institutions and individuals unique ids and associate explicit access rights to them based on the purpose of use of data. *The Access Right Policy Enforcement Point (APEP)* applies the rules to requests of *M* and

R (stored on the CHINO data store or on the organizations' data stores). Access rights on *R* and *M* can be defined on record types or single instances. If a rule is defined on a type for a specific consumer (*C*) then it can access the entire set of records of that type (i.e. the governance can be enabled to access to all *R* signaling a request for a specific service or appearing of a particular disease). A *C* that is allowed to access to *R* can automatically access also *M* about the same record type or instance. An *Ar* tuple has the following structure:

$$Ar = (O_{id}, C_{id}, R_t, R_{id}, cr_t, exp_t, Pu_{id}, FR_{id}). \tag{3.1}$$

while an instance of *Ar* tuple is shown here:

$$(1, 2, -, 41, 2013 - 1 - 1T14 : 01 : 01, 2013 - 4 - 3T14 : 01 : 01, consult., 324) \tag{3.2}$$

where O_{id} is the owner id. C_{id} (2 in the example) is the authorized consumer id. R_t is the R type that C is allowed to access. O can authorize C to access to R types or single instances (i.e. R_{id}). As it is shown in the example, the access is authorized only for the record R_{id}=41 (the R_t value is *null*). If R_t is specified then R_{id} needs to be null. cr_t and exp_t are the creation time and the expiration time of the *Ar*. Pu_{id} is the purpose of access that is allowed for that specific *Ar* (*consultation* in this case). FR_{id} (324) is the id of the filtering rule (detailed later) that is applied to that *Ar* for the purpose consultation. FR_{id} can be also *null* if no filtering rules need to be applied for the specified purpose of use of data. *Data Filtering Rules (Fr)* extend the access rights providing a fine-grained data filtering mechanism for XML or HL7 data [11]. *Fr* is defined by data owners and specifies who can access which parts of data for which purpose (e.g. to deny personal identifiable information when a record is accessed for business intelligence purpose). We use the XACML [124] language, an XML based standard, to specify filtering rules. For more details about privacy policies see Chapter 4 and in particular Section 4.4, while for a deeper analysis of access control aspects in data integration scenarios see Chapter 5.

An example of policy (simplified for readability reasons) is shown in Figure 3.4 and has been defined according to the record schema shown in Figure 3.3. According to the XACML notation, a policy specifies which actions a certain subject can perform on a specific resource while fulfilling some obligations (e.g. specific constrains) [135]. In CHINO, an action corresponds to a purpose of use of data that is business intelligence (BI) in the example policy in Figure 3.4. The set of obligations specifies which attributes the requester is allowed to access. In the example due to privacy constrains only record creation time (*crT*), patient gender (*pi/-gender*), birth date and birthplace are accessible. The *Data Filtering Policy Enforcement Point (FPEP)* applies XACML rules on *R* and *M* requests. *R* that is stored on the CHINO record

```
<Policy>
  <Rule RuleId="324">
      <ResourceMatch>ConsultationRequest</ResourceMatch>
      <Actions>
        <ActionMatch>BusinessIntelligence</ActionMatch>
      </Actions>
      <Obligations>
        <Obbligation>
            <AttributeId>recordTimestamp</AttributeId>
            <AttributeId>pi/gender</AttributeId>
            <AttributeId>pi/birthDate </AttributeId>
            <AttributeId>pi/birthPlace </AttributeId>
        </Obbligation>
   </Obligations>
  </Rule>
</Policy>
```

Figure 3.4: An example of XACML policy for the ConsultationRequest record type.

store is encrypted. To enforce filtering rules, the system needs to decrypt the records, apply the policies and then re-encrypt them before sending to the requester. If the records are stored on remote record stores, the policies can be applied at decentralized *FPEP* and then sent to the destination through CHINO. The process based approach allows these different configurations to be modelled and executed depending on data owner's requirements. To access to data and rules we identify a set of operations through which external applications interact with CHINO.

3.4.3 Data Management Interfaces (DMI)

The DMI has been defined to provide a set of CRUD (create, read, update and delete) operations over data and rules. In particular we designed DMI to manage metadata, records and Ar rules. The Fr rules need to be designed starting from the metadata and record content schema and thus require specific user interfaces that are currently left as future work. However, in our previous work we proposed a prototype implementation of the required graphical user interface [11]. The identified DMI are listed in Table 3.2 along with their input and output parameters.

Following our methodology, at the Step 4, the DMI internal business logic needs to be implemented through a business process. Next subsection describes the modelling elements that CHINO modelling framework provides and it shows how they execute operations on data and rules.

DMI	Description	Input	Output
PushRecord	Encrypts and stores R on the record store. It returns the generated unique	R_{id}. R	R_{id},
GetRecord	Returns R if the request is authorized or a denied status otherwise. It can return also a wait status to signal pending-approval or an error message if something went wrong during its execution.	C_{id}, R_{id}, Pu_{id}	$Resp_c$, R
DeleteRecord	Deletes the specified R from the record store.	R_{id}	Status code.
PushMetadata	Stores M on the metadata registry and returns the generated unique.	M_{id} M	M_{id}
SearchMetadata	Returns M that matches the searching parameters and that the requester is authorized to access	Search params.	Array of M
DeleteMetadata	Deletes a M from the metadata registry.	M_{id}	Status code.
AskForAR	Sends a message to the R owner asks for access rights for a R for a specified purpose.	R_{id}, Pu_{id}	Status code.
GrantRecordAR and **GrantMetadataAR**	Grant access rights to a consumer (C_{id}) for R or M type or instance for a specified period of time and a purpose (Pu_{id}). Optionally a filtering policy (FR_{id}) can be specified.	C_{id}, $R(M)_t$, $R(M)_id$, exp_t, Pu_{id}, FR_{id}	Status code.
RevokeRecordAR and **RevokeMetadataAR**	Revoke previously granted access rights to a consumer (C_{id}) for R or M type or instance.	C_{id}, $R(M)_t$, $R(M)_id$, Pu_{id}	Status code.

Table 3.2: List of identified DMI and their input and output parameters.

Custom Elements	**Description**	**Input**	**Output**
C1	Checks the hash (integrity) of a record.	R	True or False
C2	Sends a wait message to the requester	O_URL, Req_{id}, R_{id}, Pu_{id}.	Status code.
C3	Sends the Request Denied answer to the requester.	Req_{id}, Reason, C_{URL}.	Status code.
C4	Sends the Timeout answer for the call to the requester.	R_{id}, C_{URL}.	Status code.
C5	Calls the AskForAR DMI.	O_{id}, C_{id}, R_{id}, Pu_{id}	Status code.
C6, C7	Grants Record and Medatada access rights calling the corresponding LLO on APEP.	C_{id}, R_t, R_{id}, exp_t, Pu_{id}, FR_{id}	Status code.
C8	Sends a reminder message to the R owner	O_{URL}, C_{id}, R_{id}, Pu_{id}.	Status code.
C9	Applies a filtering rule Fr on R.	R, O_{id}, Pu_{id}	Filtered R
C10, C11	Performs a call to LLO to check the AR for a given consumer (C_{id}) and purpose (Pu_{id})	C_{id}, R_{id}, Pu_{id}	True or False
C12	Encrypts R	R_{id}	Status code.
C13	Customizable logging on internal or external auditing system.	Process state info	Status code
C14	Saves M on metadata registry internal components.	M_{id}, M	Status code
C15	Saves R on record store or internal components.	R_{id}, R	Status code
C16	Decrypts R	R_{id}	R
C17	Sends the requested R to the requester.	R, C_URL	Status code.
C18	Restore the requested M from metadata registry.	M_{id}	M
C19	Restore the requested R from record store.	R_{id}	R
C20	Waits for the approval of an access right request.		
C21	Sends an error message to the requester.	Req_{id}, $Reason.$	Status code.
C22	Deletes M from metadata registry	M_{id}	Status code.
C23	Deletes R from record store	R_{id}	Status code.

Table 3.3: List of the CHINO custom modelling elements and their input and Output parameters.

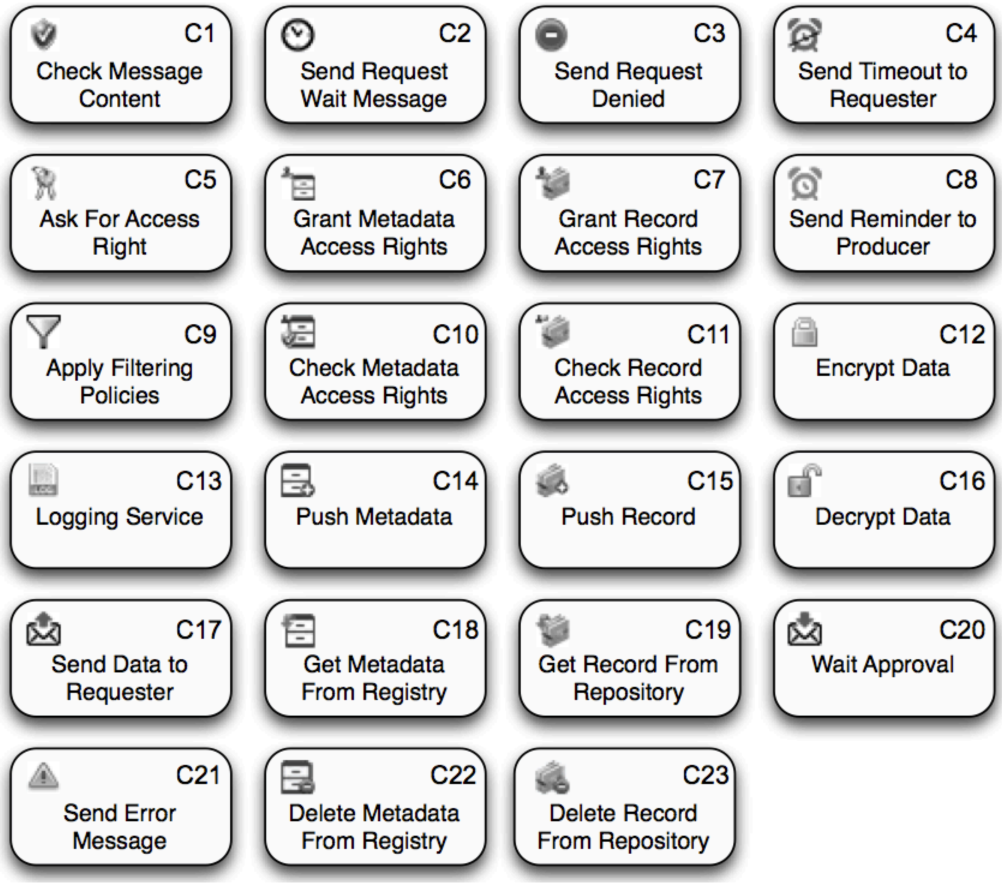

C1
Check Message Content
C2
Send Request Wait Message
C3
Send Request Denied
C4
Send Timeout to Requester
C5
Ask For Access Right
C6
Grant Metadata Access Rights
C7
Grant Record Access Rights
C8
Send Reminder to Producer
C9
Apply Filtering Policies
C10
Check Metadata Access Rights
C11
Check Record Access Rights
C12
Encrypt Data
C13
Logging Service
C14
Push Metadata
C15
Push Record
C16
Decrypt Data
C17
Send Data to Requester
C18
Get Metadata From Registry
C19
Get Record From Repository
C20
Wait Approval
C21
Send Error Message
C22
Delete Metadata From Registry
C23
Delete Record From Repository

Figure 3.5: The CHINO custom modelling elements.

3.4.4 Modelling Elements

To provide a comprehensive set of elements and facilitate the business process modelling activity, the CHINO modelling framework borrows the whole set of standard BPMN 2.0 elements [139] and in addition, it identifies a set of custom modelling elements that are shown in Figure 3.5 and detailed in Table 3.3.

BPMN has been preferred to other languages that provide visual representations due to its maturity and flexibility. In particular, BPMN version 2.0 allows users to define orchestrations among systems and people and many reference implementations of execution engines have been proposed. To facilitate operations on data and rules through calls on the Low-Level Operations (LLO), we introduce a set of custom elements. In addition to operations on data and rules, some custom elements implement the DMI logic (i.e. send record or error messages to requesters). These operations are performed in a transparent way to the analysts and developers that will need simply to drag and drop and configure the elements. Process model examples implementing the GetRecord DMI are shown in Section 6. The set of custom elements has been defined based on identified requirements and policies in Table 3.1. Thus it could happen that new elements will be required to manage exceptional cases. In that case, BPMN standard elements can be easily customized and added to the list of custom elements if considered as potentially useful in future situations. The modelling elements access to internal components that manage data and rules through a uniform set of interfaces called LLO

3.4.5 Low-Level Operations (LLO)

The identified operations on data and rules are called Low-Level Operations (LLO) and are executed by the internal CHINO components. These components include the record store, metadata registry, and policy enforcement points. As for the DMI, the set of LLO has been defined based on the CRUD operations on data and rules that are performed by modelling elements. Since LLO have been designed following the same approach as for the DMI reflecting the custom modelling element operations, we do not list them here.

Next section will first explain the relation between policies and the DMI operations and then it will show how the data, rules, DMI, LLO and modelling elements are used to enforce the identified policies.

3.5 Policies and Policy Enforcement Templates

The *Definition of Executable Processes and Policies* step of the CHINO methodology starts with the identification of the right sequence of calls to the DMI operations that respect the high-level interaction requirements identified at Step 3. Namely, each step in Figure 3.1 can be seen as one or many calls to DMI. For each DMI, business analysts need to identify which policies need to be satisfied during its execution. Then, they need to define, through the combination of data, rules, DMI, LLO and modelling elements, the DMI internal business logic in such a way that it satisfies the identified policies. Such definition will result in compliant DMI business logic and therefore in compliant data sharing among involved stakeholders.

To give guidelines and facilitate the process definition, we propose a set of process enforcement templates, each of them enforcing partially or totally some of the identified policies. Before showing how the processes are defined with respect to identified policies and enforcement templates, we show how doctor, specialist and patient interact in the doctor-consultation scenario underlying how for each single interaction certain policies need to be enforced.

3.5.1 Doctor-Consultation with CHINO

The sequence of message exchanges among actors involved in the doctor-consultation scenario according to the OpenMRS and CHINO integration is shown in Figure 3.6. Here we focus on the interaction steps while the architectural and integration details are shown later in Sections 7 and 8.

The sequence starts with the doctor filling the consultation request on his OpenMRS instance and sending it to the specialist through the CHINO platform. This involves three operations. (1) A record C_R is generated and sent with a call of $PushRecord(C_R)$ operation. (2) The corresponding instance of metadata C_M is sent to CHINO metadata registry through $PushMetadata(C_M)$. (3) $GrantRecordAR(C_R,\ S,\ exp_t,\ consultation)$ grants access rights to the specialist (S) to access C_R for a period of time until exp_t for purpose consultation. (4) The specialist will find the C_M through a $SearchMetadata(C_t)$ query where C_t is the consultation request type. The searching can be done cyclically at fixed period of time or each time the specialist logs into his system. The developed OpenMRS integration module implements the later case. Then, (5) S asks and obtains the C_R performing a $GetRecord(S,\ C_R,\ consultation)$. To understand better the patient health status, at (6) the specialist asks for more patients' data requesting records $r1$, $r2$ and $r3$. Assuming S does not have Ar to access $r2$, then at (7) a message $AskForAccessRight(S,\ r2,\ consultation)$ is sent to the record data owner that in this case corresponds to the patient. Currently the *Ask For Access Rights* custom element sends an email with two links carrying the $correlation_{id}$ that will allow later CHINO to resume the process. One link approves while the other denies the request. The patient at (8), clicking on the approval link, performs a $GrantRecordAR(r2,\ S,\ exp_t,\ consultation,\ correlation_{id})$ that will grant the Ar for $r2$. At (9), finally, S will be able to access to $r2$ through a $GetRecord(S,$ $r2,\ consultation)$. Once filled the consultation response, S sends the corresponding R and M through steps (10), (11), (12). Finally (13) the doctor can get the response.

The overall sequence of steps allows the involved actors to exchange data through the identified set of DMI. We now analyse the relation between the DMI calls and the policies that need to be enforced during their execution. We then propose a set of policy enforcement templates that could be used to implement the DMI process models while enforcing each of the identified policies.

3.5.2 DMI Policies

Once the business analysts have identified the right sequence of DMI calls for a scenario such as doctor-consultation, the *Definition of Executable Processes and Policies* step is to identify for each DMI operation the set of policies that need to be enforced during its execution. If we consider the *GetRecord* operation, the set of policies includes *Identification*, *Access*, *Purpose*

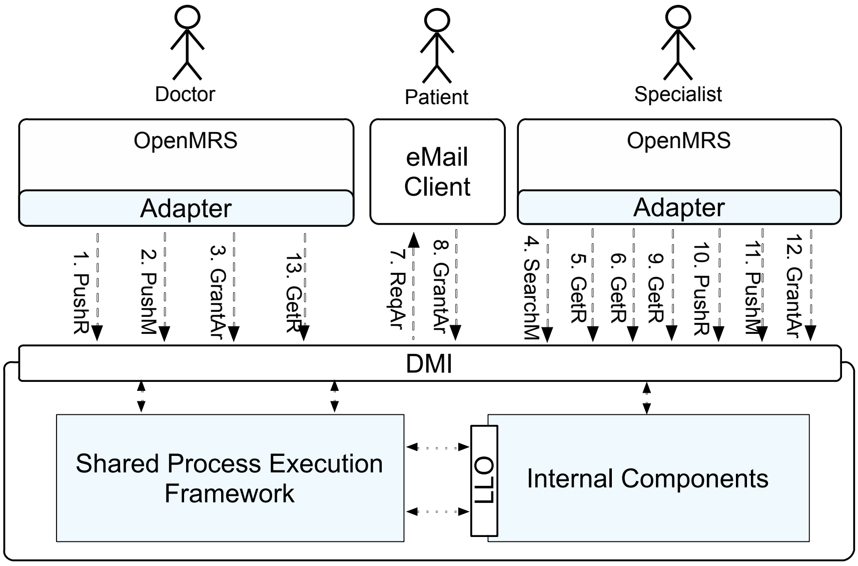


Figure 3.6: The sequence of steps carried out by the doctor, the patient and the specialist during the doctor-consultation scenario.

and *Proportionality*. In other words, each time a record is requested, the requester needs to be identified and the access to data needs to be authorized for the specified purpose. Finally, the requester needs to obtain only the necessary data to perform the tasks related to the specified purpose. Policies can be either extracted from regulations or the set of organization specific requirements during the first 3 steps of our methodology.

The proposed methodology identifies the actors and the steps that need to be performed in order to identify and formulate the exact set of policies that each DMI needs to enforce. During our regulatory analysis we identified some state-of-the-art tools and methods that can be used by business analysts during the policy identification to devise the set of policies affecting each of the DMI's [33, 143, 122, 159].

Following our approach, the policies are enforced with the appropriate implementation of the DMI process models. It is important to note that some of the policies (e.g. *Identification* or *Safeguard*) are satisfied by design with the appropriate design choices and implementation of internal components (e.g. encryption strategy on record store). Therefore these policies are not considered during process modelling but instead are they assumed to be supported by the data management framework itself.

If an organization (i.e. its business analysts) can define for each DMI a process model that enforces the identified policies, that organization is compliant with its regulations relatively to the data management operations provided by the CHINO framework. Note that herein the compliance refers to the data management operations while the whole regulatory compliance

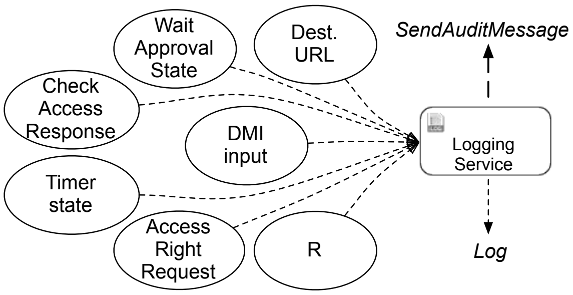


Figure 3.7: CHINO Logging Service custom modelling element sending logging messages to a specified auditing system.

involves also other requirements not strictly related to data management, such as the establishment of supervisory authority or other administrative obligations.

The definition of process models can represent a complex task. To facilitate the process modelling activity and teach to users how to use the CHINO modelling framework, we provide a set of policy enforcement templates that aim at helping users in defining compliant process models.

3.5.3 Policy Enforcement Templates

To show how the identified policies are enforced within the CHINO framework we propose a set of policy enforcement templates. These templates are reusable and specialized BPMN process design patterns [175] that aim at providing a set of guidelines, best practices and teaching to users how to enforce the identified policies using the CHINO modelling framework. In particular, these templates show the usage of CHINO modelling elements and how they interact with internal components to achieve policy enforcement.

The next subsections will detail the identified templates describing for each of them the policies it enforces (partially or totally), input and output parameters and the modelling elements that need to be used.

Auditing

One of the most important requirements for achieving compliance in healthcare is represented by the need for ensuring effective auditing strategies [159]. If an organization needs to feed personal auditing systems, the Logging Service can be customized to format and forward the audit messages to the external audit services. The audit message is composed using Groovy Scripting language (groovy.codehaus.org) that is one of the scripting languages offered by the BPMN engine.

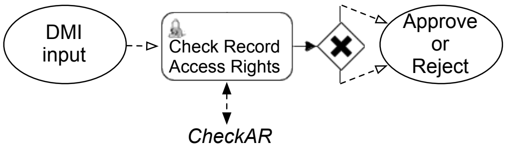


Figure 3.8: Template showing the usage of the Check Record Access Right CHINO custom modelling element.

Figure 3.7 shows the Auditing template and the Logging Service usage. All the attributes stored inside process space can be used as input parameters to compose the audit message. The message can be either sent to external audit systems or logged internally calling the Log LLO. It is important to note that the availability of the set of input parameters depends on the elements that are executed before the *Logging Service*. For example immediately after the process starts, only DMI Input parameters are available to compose audit message.

Check Access Rights

Security policies have as objective to ensure access control and secure data management (storage and transfer). While policies such as *Identification* and *Safeguard* do not necessarily require modelling elements to be enforced, the *Access* and *Purpose* policies are ensured by the *Access Right Policy Enforcement Point* component.

Figure 3.8 shows the usage of the *Check Record Access Right* modelling element. It takes in input the DMI parameters and returns either approve or reject response. It is shown in conjunction with a BPMN Exclusive Gateway to show how the output is used to take the appropriate decision afterwards.

Ask for Approval

When the data owner is an external entity (e.g. patient), to enforce the *Consent* policy, the *Ask For Access Rights* element needs to be used. To signal to the requester that the process is waiting for the approval, a wait message needs to be sent as the DMI call response.

The template in Figure 3.9 shows parallel execution of the *Ask For Access Right* and the *Send Request Wait Message* CHINO modelling elements to send an access request to the owner and a wait message to the requester. The proposed template is usually followed by a sequence of steps to wait for approval.

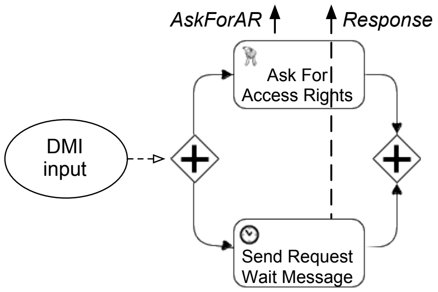


Figure 3.9: Template showing how *Ask For Access Rights* should be paired with the *Send Wait Message* to requester.

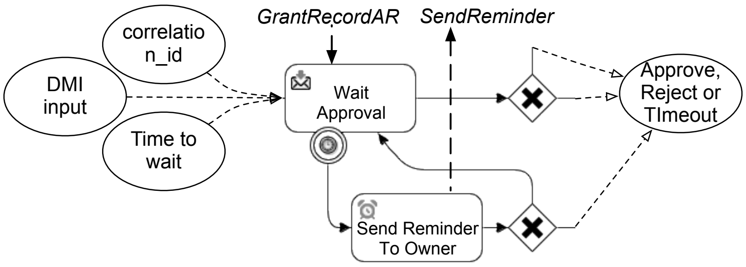


Figure 3.10: Combination of CHINO custom and BPMN modelling elements to ask to an external data owner to approve a record request.

Wait for Approval

The Ask for Approval template is usually followed by a sequence of steps that need to implement the waiting for approval logic. The waiting logic depends on the data owner requirements.

The template in Figure 3.10 uses the *Wait Approval* CHINO modelling element that takes in input of the $correlation_{id}$ and the DMI input. The time to wait is used inside the BPMN Boundary Timer element to trigger periodically the execution of Send Reminder to Owner element. The output can be either a request approved, a request rejected or time-out response. The given output needs to be sent later as the response to the data requester.

Retrieve Record

The *Retrieve Record* template shows the usage of the Get Record From Repository modelling element that restores a record from a local or remote record store.

The template is shown in Figure 3.11 and is used to enforce the *PrivateStore* policy that organizations (e.g. the one under Italian jurisdiction) can use to manage their records inside their information systems while participating to CHINO sharing platform. The template takes

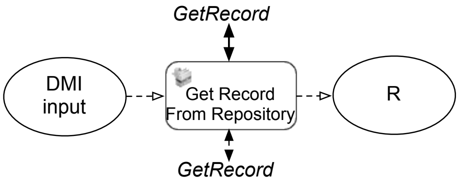


Figure 3.11: Template showing Get Record From Repository element configuration to retrieve a record from local or remote record store.

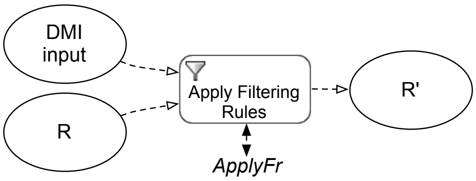


Figure 3.12: CHINO custom modelling element applying filtering rules to delete unnecessary data for the specified purpose.

in input being the DMI input parameters and returns the retrieved R.

Apply Filtering Rules

As already discussed in previous sections, the Proportionality policy is satisfied by applying Fr inside the *Apply Filtering Policies* CHINO modelling element that calls the *Policy Filtering Enforcement Point* internal component.

The enforcement template in Figure 3.12 shows the *Apply Filtering Policies* modelling element usage. It takes in input being the DMI input parameters and the retrieved R. Once applied Fr by calling the *ApplyFr* LLO, it returns the filtered R' to the requester.

Return Response

Each DMI is an asynchronous operation. Once request data is transmitted to the DMI endpoints, the connection is terminated with an HTTP success status. While some calls (e.g. $GrantRecordAR$) do not necessary require a response, for other DMI (e.g. $GetRecord$) the response is mandatory. To reply to $GetRecord$ operation, CHINO provides four types of responses implemented by corresponding custom modelling elements. *Send Record to Requester*, *Send Request Wait Message*, *Send Request Denied*, *Send Error Message* are the elements that need to be used to provide responses. Examples of these elements are shown in Figures 3.13 and 3.14 explained later.

Exception Handling

BPMN offers a set of modelling elements (e.g. *ErrorBoundaryEvent*) that can be used to catch exceptions at the process model level and implement the corresponding corrective logic. When some exceptions (e.g. broken connection with remote store) are caught and proper actions taken, a response signalling an error may be sent to the requester. In this case we provide the *Send Error Message* modelling element part of the previously shown *Return Response* template. An example of the usage of the *Send Error Message* element to signal to the requester that an error happened is shown in Figure 3.14 at step i11.

Data Representation

Organization specific requirements are related to customization aspects and their enforcement facilitates organization participation to EHR programs. For example organizations may want to use existing systems that encode data in a proprietary format. In order to exchange data within EHR programs and with other organizations they need to implement transcoding procedures. To solve these challenges we do not propose a specific template but leave the suggestion that within the BPMN Service Tasks transformations can be implemented to transcode data before releasing it to third party. Java or Groovy languages can be used inside our framework to perform such tasks.

Other templates

To enforce all security and privacy policies CHINO custom modelling elements play a central role. With high level of customization offered by the CHINO modelling framework, different entities can be identified and being involved in the policy granting right process (as shown by the *Ask for Approval* template). When *Ask for Approval* is used properly in conjunction with the *Auditing* template they are able to enforce the *Accountability* policy. In fact, accountability is defined as ensuring the access and auditing of all operations over data. The *Integrity* policy is achieved by verifying data integrity upon retrieval and transmission and using the *Check Hash* custom CHINO element. The process-based approach is able to manage easily the exceptional cases in which data subjects are under certain age or the records are about mental problems and should not be disclosed to the subjects.

If the proposed set of custom modelling elements and templates is not sufficient to implement some special requirements and exceptional cases, the standard BPMN elements can be used and extended with the help of system administrators.

With the presented templates we show how the identified policies in Section 2 are satisfied through appropriate combination of modelling elements described in Section 4. Such templates will be used later to compose the whole DMI process models.

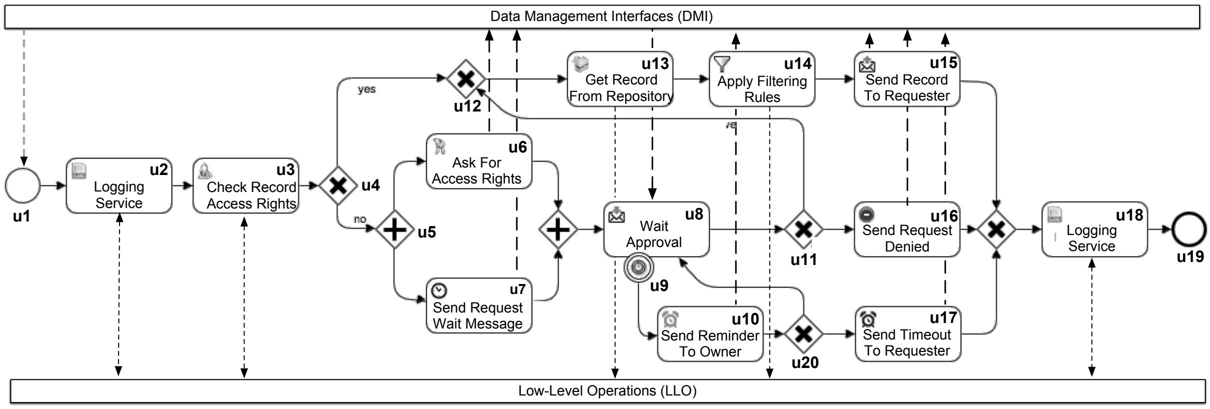


Figure 3.13: A business process implementing the GetRecord DMI according to UK regulations and standards.

3.6 CHINO Process Definition and Execution

In Section 3.4 we identified the elements that characterize the CHINO modelling framework and in Section 3.5 we clarified the relation between policies and DMI. Then we proposed the policy enforcement templates that can be combined to enforce DMI policies and define compliant processes. Here we show examples of such processes and how they are defined and executed inside the CHINO process execution framework.

3.6.1 Process Definition

To show how process models can be defined through the combination of proposed templates and other modelling elements we analyse the *GetRecord* operation implementation according to the identified Italian and UK regulatory policies. It first needs to check access rights, retrieve requested record from the repository and return it to the data requester.

As shown previously, some of the policies that the *GetRecord* needs to enforce are *Identification*, *Access*, *Purpose* and *Proportionality*.

Figure 3.13 shows the GetRecord process model (a simplified version for readability reasons) according to UK regulations while Figure 3.14 shows the process compliant with Italian regulations and national-wide EHR architecture [143]. The processes show differences in the use of modelling elements and policy enforcement templates due to different requirements. Namely, both processes start with a BPMN Start Event element $u1$ and $i1$ respectively.

Then the *Check Access Right* template is used in both processes to authorize the request. In Figure 3.13, the rule checking is followed by the *Ask for Approval* and *Wait for Approval* templates that collectively enforce the UK *Consent* policy. In case of access right positive

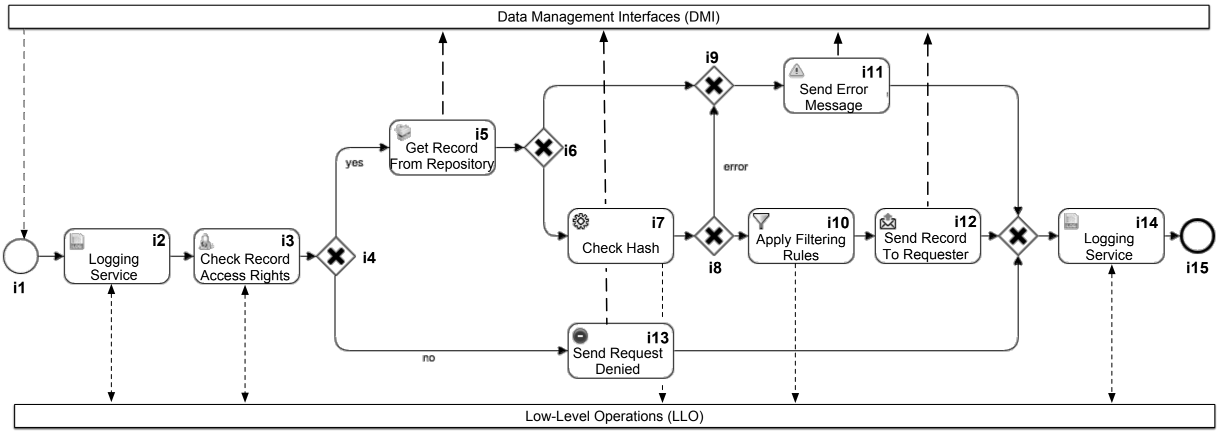


Figure 3.14: A business process implementing GetRecord DMI according to Italian regulations and standards.

result, the *Retrieve Record* template (elements $u13$ and $i5$) retrieves the requested record. It restores the record from local record store in $u13$ or remote store at $i5$ as defined by the Italian regulations, which is subject to the *PrivateStore* policy. The *Return Response* elements $u15$, $u16$ and $u17$ and $i11$ are used to send the four type of return result for the *GetRecord*. Namely, $u15$ returns the requested record, $u16$ the request denied response, $u17$ a time-out and finally the $i11$ returns an error message. Since in both Italy and UK regulations the *Proportionality* policy needs to be enforced, the *Apply Filtering Rules* template (elements $u15$ and $i11$) apply the Fr before returning the filtered R to the requester.

As we shown with these examples, the combination of modelling elements, data, operations and rules provides a comprehensive set of constructs that can implement the DMI internal business logic while enforcing identified policies.

3.6.2 Data Management Process Execution

The resulting CHINO process models are fully compliant with BPMN 2.0 processes where the custom elements are BPMN *Service Tasks*. *Service Tasks* have been introduced to provide powerful extension points using custom code (Java in our case) to perform custom actions (e.g. calls of web services). This feature is offered by almost all BPMN engines that allow the implementation of the *Service Task* business logic through programming language interfaces (i.e. Java Interfaces). The CHINO custom elements are *Service Tasks* that have been modified introducing a new look and feel, additional configuration parameters and a Java class for each of them to implement the calls to the LLO and interaction with external systems. This BPMN customization approach has multiple advantages. First, these extensions do not affect the process execution semantic and therefore there is no need for process engine modifications. Thus, the CHINO processes can be easily ported to a different BPMN engine that supports Java language and *Service Task* extensions. Secondly, modelling of such CHINO processes does not introduce the need for specialized skills; instead it hides technical implementation details while allowing users to fully customize the processes.

Before being deployed, the processes need to be manually verified in order to identify possible syntactic errors. Once verified, the processes are deployed and executed on the process engine. The engine manages process persistence, their starting and ending, and concurrent execution. An important aspect is related to the fault tolerance that includes exception handling, process cancellation and process restoring in case of errors or system crashes. Exceptions that are related to process model aspects can be caught by following the general guidelines proposed by the Exception Handling template. Exceptions that are related to other errors such as system crashes need to be managed by the system administrators. An administrator manual intervention will be required to restore the correct process state and restart it. The process restarting and roll-back procedure depends on the BPMN engine. The engine adopted by CHINO (detailed later) does not provide automatic roll-back functionalities that would need to be implemented additionally. Instead, it provides the access to the database and possibility to modify and update process information and restart it.

One important aspect that is worth mentioning, but left as future work, is the detection of potential policy conflicts. Policy conflicts are one of the major issues of policy-based approaches in inter- and intra-organization collaboration scenarios. It has been widely studied and many solutions for the conflict-aware policies and processes definition have been proposed [112]. Currently, we assume that policy conflict detection and resolution are done at design time. In particular major conflicts could arise in case of wrong process definition. Namely, since organizations will need to define one process for each DMI, the conflicts could arise relatively to the authorization aspects [112] if processes are not implemented correctly to grant the access rights and retrieve records. For example, granting access rights and retrieving records need to be defined in such a way that their combination results in a correct access right granting and data sharing. These aspects are currently left as future work.

Next section shows more technical details about the CHINO prototype architecture and implementation.

3.7 CHINO Architecture and Implementation

This section details the system architecture that supports the CHINO common execution framework. We also present implementation details and describe how processes are defined and deployed using the modelling framework.

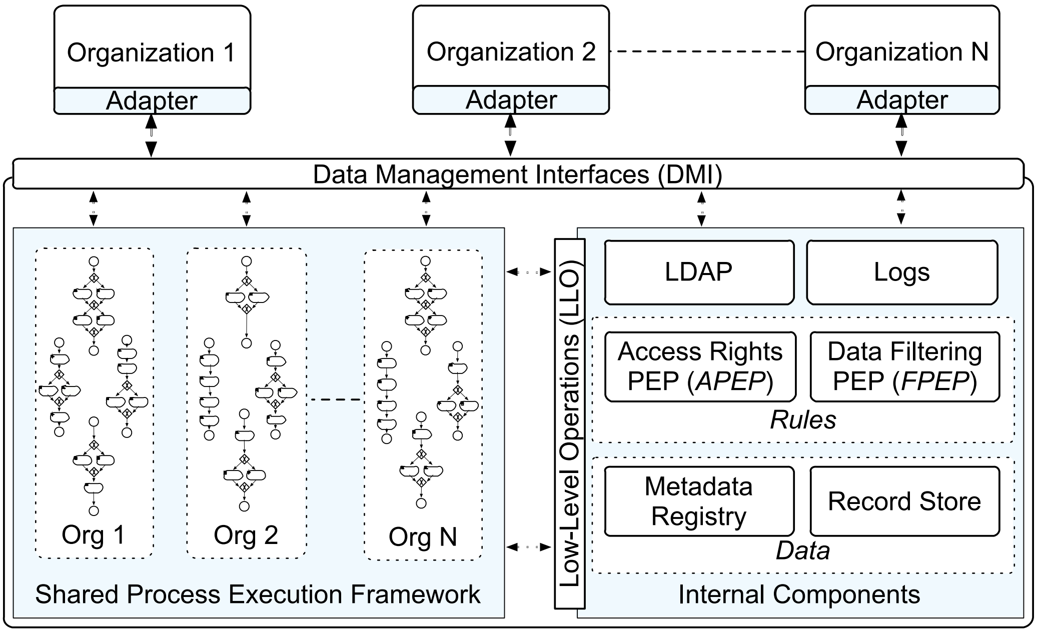


Figure 3.15: The conceptual architecture of the CHINO framework.

3.7.1 The Common Execution Framework

Following the CHINO methodology, once the processes and rules are defined (*Step 4*), they are deployed and executed inside the shared execution environment (*Step 5*).

Figure 3.15 shows the CHINO conceptual architecture, which is based on the combination of BPM and SOA/EDA architectural patterns, in order to provide clear separation between business level logic and technical integration details [6, 68].

BPMN serves for modelling and executing business processes and orchestrating all the involved actors (e.g. people, external systems and internal components). Service-Oriented (SOA) and Event-Driven (EDA) architectural patterns and technological components (e.g. ESB) facilitate the creation of both synchronous and asynchronous APIs to interact with external entities and internal components (e.g. DMI and LLO). These components are responsible for technical issues such as reliability, scalability, and communication with potentially heterogeneous information systems using different communication protocols. The business-level integration supported by the BPM framework deals with a higher level of integration by enforcing the policies and requirements of each participant organization.

In Figure 3.15 the shared process execution framework is shown on the bottom left-hand side. It executes and stores the business processes that perform calls on LLO and implement the DMI business logic. Each organization has its own set of DMI processes that can be seen as logically grouped and belonging to it and that are stored on the shared process repository.

Process templates can be also shared among organizations enabling their reuse. If a process satisfies the requirements of a new organization, from a template, a private copy of the process can be created and customized. The process does not need to be modified if only access rights and filtering rules need to be changed.

The set of internal IT components is shown at the bottom right-hand side. The core internal components can include the data containers such as the shared *Record Store* that stores medical records and the *Metadata Registry* that stores the metadata records. Then, the *APEP* manages the access rights while *FPEP* applies fine-grained purpose-based data filtering rules.

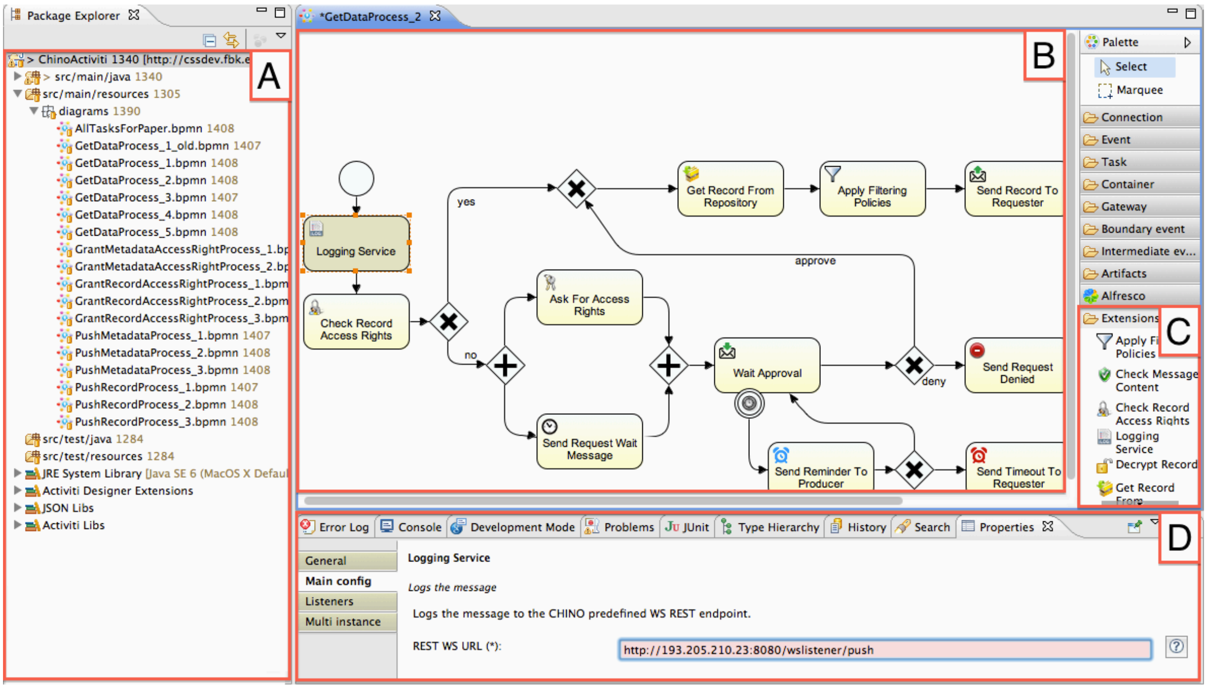
The proposed architecture has been tested and validated with the development of the CHINO prototype. The prototype and its integration with OpenMRS, demonstrate that the proposed architecture efficiency is comparable with classical SOA architectures and design patterns [6]. In particular, one aspect that is worth mentioning is the fact that the introduction of BPM technology at the internal logical integration layer does not introduce additional limitations. This is possible due to the asynchronous definition of the DMIs to implement the messaging between CHINO and participant organizations. In such way, the combination of SOA and EDA technologies enables the processes long-lasting execution without introducing additional challenges. As next we show implementation details of the designed framework.

3.7.2 The Prototype Implementation

Building such a framework that supports the whole CHINO methodology and implements the proposed architecture is a challenging task. The implementation provides a seamless integration between all the framework components in order to address the general objective, that is, to provide a common execution framework that is able to orchestrate data sharing services, business process management, heterogeneous communication protocols, data management components and distribution of policy enforcement points.

As the shared process execution framework, we adopt Activiti BPMN, an open source BPMN 2.0 process engine [2]. It offers a process designer plugin for the Eclipse IDE (eclipse.org) and a process engine and repository that stores and executes the business processes deployments. In terms of runtime support, Activiti provides long-term persistence of processes and concurrent process execution. It also provides the process correlation mechanism linking the DMI calls with the appropriate running process instances using the $correlation_{id}$ that is carried within messages. External systems interact with the deployed processes through calls of DMI implemented by the underlying Enterprise Service Bus (ESB).

As the ESB we use an open source Java-based ESB called Mule [126]. Mule lies at the heart of prototype implementation orchestrating all the frameworks' components. It exposes all the interfaces to external information systems, communicates with Activiti, and it hosts the data management components exposing them through LLO. The ESB provides message persistence,


Package Explorer
> ChinoActiviti 1340 [http://cssdev.fbk.e
> src/main/java 1340
src/main/resources 1305
diagrams 1390
AllTasksForPaper.bpmn 1408
GetDataProcess_1_old.bpmn 1407
GetDataProcess_1.bpmn 1408
GetDataProcess_2.bpmn 1408
GetDataProcess_3.bpmn 1407
GetDataProcess_4.bpmn 1408
GetDataProcess_5.bpmn 1408
GrantMetadataAccessRightProcess_1.bp
GrantMetadataAccessRightProcess_2.bp
GrantRecordAccessRightProcess_1.bpm
GrantRecordAccessRightProcess_2.bpm
GrantRecordAccessRightProcess_3.bpm
PushMetadataProcess_1.bpmn 1407
PushMetadataProcess_2.bpmn 1408
PushMetadataProcess_3.bpmn 1408
PushRecordProcess_1.bpmn 1407
PushRecordProcess_2.bpmn 1408
PushRecordProcess_3.bpmn 1408
src/test/java 1284
src/test/resources 1284
JRE System Library [Java SE 6 (MacOS X Defaul
Activiti Designer Extensions
JSON Libs
Activiti Libs
A
*GetDataProcess_2
B
Logging Service
Check Record Access Rights
yes
no
Get Record From Repository
Apply Filtering Policies
Send Record To Requester
Ask For Access Rights
Send Request Wait Message
Wait Approval
approve
deny
Send Request Denied
Send Reminder To Producer
Send Timeout To Requester
Palette
Select
Marquee
Connection
Event
Task
Container
Gateway
Boundary event
Intermediate ev...
Artifacts
Alfresco
Extensions
C
Apply Fi Policies
Check Message Content
Check Record Access Rights
Logging Service
Decrypt Record
Get Record From
Error Log
Console
Development Mode
Problems
JUnit
Type Hierarchy
History
Search
Properties
D
General
Main config
Listeners
Multi instance

Logging Service

Logs the message

Logs the message to the CHINO predefined WS REST endpoint.

REST WS URL (*): http://193.205.210.23:8080/wslistener/push

Figure 3.16: A screenshot of the CHINO policy and process modelling framework based on the Activiti process designer.

transaction management and many other technical features. Mule provides integration with Activiti runtime through an embedded (Spring based) or standalone (REST) fashion [2].

The DMI and LLO are exposed as HTTP endpoints using JavaScript Object Notation (JSON), a lightweight data exchange format [51]. A Lightweight Directory Access Protocol (LDAP) server ensures authentication for DMI calls. Mule ESB uses the Spring security framework [162] to access to the authentication and security layer of LDAP. The underlying data model is deployed on MySQL database (mysql.com) and it is used by all internal components (i.e. *Metadata Registry, Record Store, APEP, FPEP and Logs*). As *Metadata Registry* we used an instance of freebXML registry (ebxmlrr.sourceforge.net), a free and open source ebXML reference implementation [34]. It has been adopted in various projects and sponsored by different institutions like IHE as the standard for metadata storage in healthcare [93]. ebXML Registry is based on a flexible data model that can store any type of metadata. *Record Store* is built as a map (R_{id}, $R_{content}$) where R_{id} is the key used to identify encrypted records ($R_{content}$). The design choices of the scalable record store have been inspired by our previous work on record management in cloud based environments. In particular we consider the work [108] as an enterprise level replacement for our *Record Store* that does not introduce delays or bottlenecks when big quantities of data need to be managed centrally (for more details see [108]). *Access*

Right PEP has been designed to store the access right rules for records and metadata. The operations on *APEP* are *GrantAR* that stores access rights, *RevokeAR* that revokes them, and *CheckAR* that verifies if an O_{id} has required access rights to access an R_{id}. *Data Filtering PEP* is implemented using the open-source project called Enterprise Java XACML Implementation [66]. The validation of the FPEP component with real case filtering policies has been performed within our previous work [9]. The development of user interfaces for the creation of policies is left as future work, while *FPEP* component and *ApplyFR LLO* that perform calls on it have been developed. As already shown, the technical details related to the access to the internal CHINO components are hidden to the final users by using the custom modelling elements. This simplifies the process definition and process understandability.

3.7.3 Process Definition and Deployment

The process-modelling and policy definition framework, as described by the methodology, involves the collaboration of the business analysts and developers. With respect to modelling, we extended the Activiti plugin for Eclipse IDE by adding the CHINO custom modelling elements. Figure 3.16 shows the editor at work. The added custom elements (described in Table 3.3) are present under the Extensions tab in Section C of Figure 3.16. Developers would need only to input some configuration parameters in the Properties tab shown in Section D such as the URL where the Logging Service will send the audit messages on the external auditing service. Process modelling is performed in Section B. The process models are deployed on the engine using Ant (apache.ant.org) deployment scripts accessible in Section A. Once deployed, the processes become automatically executable to manage organizations data. The process management is performed by the ESB that will dynamically start or resume the execution of the specific caller organization. To link a process with an organization we use the *ProcessKey* which identifies univocally a process model. Namely each process key is named as a pair of *DMI* name and O_{id} (i.e. DMI-O_{id}). Once named and deployed the processes are automatically triggered by calls to *DMI* performed by the identified organization.

To test the CHINO platform and the defined processes and policies we integrated it with OpenMRS.

3.8 Validation

Here we report how we validated CHINO from different perspectives. First we validated its technical ability to execute data sharing processes according to the identified requirements and policies. Then we validated the CHINO modelling framework usability to verify if business process modellers are able to define CHINO processes based on identified requirements. Finally, together with privacy experts we analysed if and how CHINO can satisfy regulatory requirements in the Italian legislation scenario.

We start by describing the first validation phase by integrating CHINO and the OpenMRS system.

3.8.1 The OpenMRS Integration

To show how the CHINO system prototype achieves data sharing across multiple organizations via the defined compliance-aware data management processes, we have conducted the integration between CHINO and an open source medical record system called OpenMRS [146], which is used widely in the world. To test the interaction we developed an OpenMRS module for the doctor-consultation scenario called *ChinoOpenMRSModule*. Then we developed two different sets of processes and policies to simulate the configuration in which the specialist operates under Italian legislation while the doctor is under UK legislation. Both OpenMRS instances rely on a common CHINO policy execution environment deployed separately from these two OpenMRS instances. The testing phase has demonstrated also that the innovative CHINO architecture does not introduce any bottleneck or inefficiency if compared to more classical architectures. The process execution in particular does not introduce significant overhead to the interactions among client applications and CHINO.

The CHINO OpenMRS Module

The module has been developed according to OpenMRS guidelines and using Spring technology [162].

Its conceptual architecture is shown in Figure 3.18. All messages exchanged between *CHINOpenMRSModule* and CHINO, are encoded as JSON messages. The module performs calls to the DMI through the WS Sender library that uses the Parser to create JSON messages according to DMI interfaces. Then it receives the requested records and all other messages through the WS Listener component. WS Listener has been implemented as a RESTful web service to receive JSON messages. Once received, the component parses them using the Parser and stores on OpenMRS internal DB. The OpenMRS data model has been extended to store message content, their type and the id. This allows the module to check later if new messages have been received. The Interceptor component is developed to catch records generated by OpenMRS (e.g. new patient registration) and to send them to CHINO. In such way the records and their metadata will form the patient medical history on CHINO and will be available for future queries by other systems. A screenshot of the doctor consultation form used by the doctor is shown in Figure 3.18. The specialist is provided with a similar UI to view the consultation requests and provide responses.

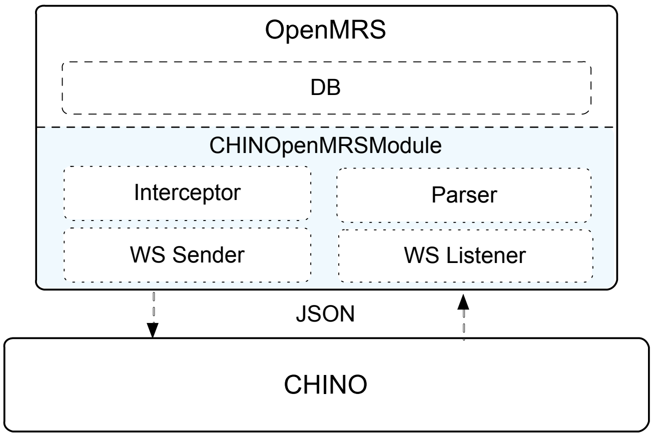


Figure 3.17: The conceptual architecture of the developed module.

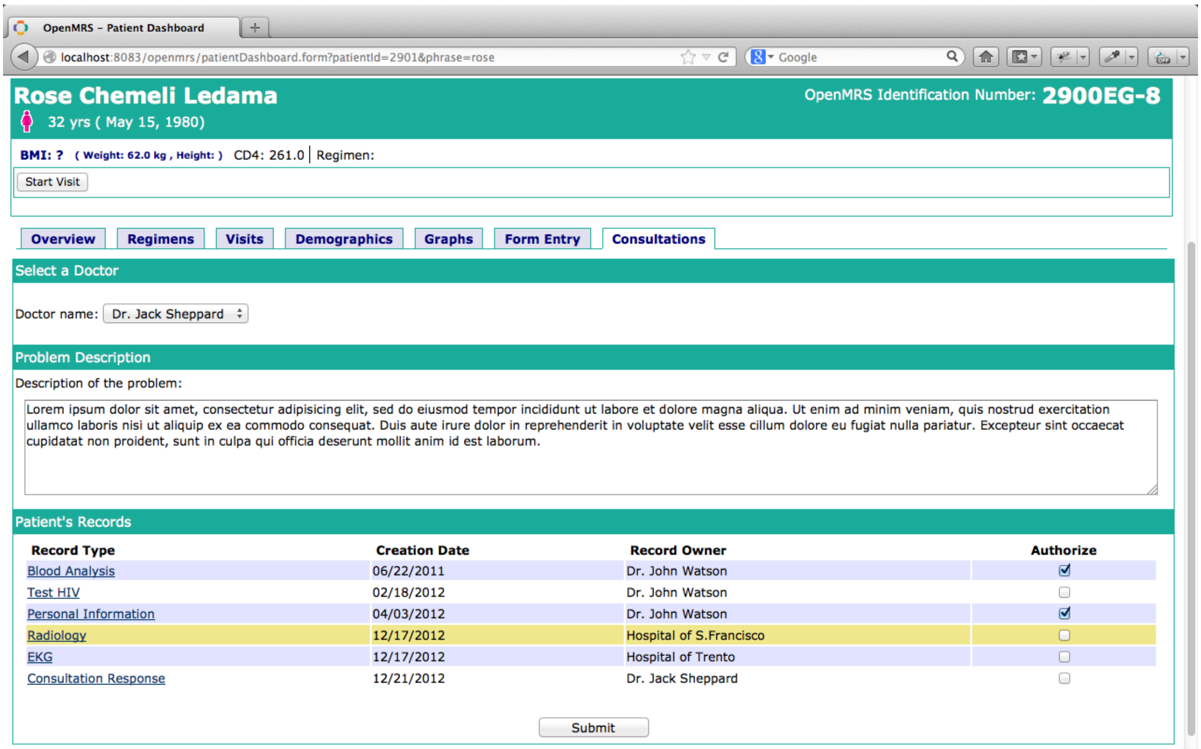
Data Exchange in Doctor-Consultation

The simulation of the doctor-consultation scenario allows us to test the CHINO framework and exchange data between the two instances. The doctor-consultation starts with the doctor filling the form (shown in Figure 3.18) on his OpenMRS instance. When doctor press the “Submit” button a consultation request CR is generated and sent to the CHINO platform. The module, using the Parser, transforms it into a JSON message and sends it to CHINO using the WS Sender, which calls the $PushRecord(C_R)$ operation. The same sequence of steps is done also for the other operation calls. When the specialist asks for the r2 performing a $GetRecord(r2)$ he receives a wait message as response. When patient approves the request with the $GrantRecordAR$ the process will be resumed and the record will be sent to the specialist’s WS Listener and saved on the database. The same sequence of steps will be performed when specialist will send the response.

As a result, we can show that one organization on an OpenMRS instance can perform data sharing with the organization on a different OpenMRS instance having different data management requirements and policies.

The developed scenario shows typical interactions among patients, doctors and institutions having different set of security and privacy requirements. We analysed the related regulations and we provided a methodology that goes from requirement collection to definition of data sharing processes and then to the their execution. With CHINO methodology and framework we show how the identified challenges can be addressed and compliance-aware data sharing supported.

We have shown that CHINO can support data exchange and interaction among medical record systems according to defined processes and policies. Next section reports our analysis


OpenMRS – Patient Dashboard
localhost:8083/openmrs/patientDashboard.form?patientId=2901&phrase=rose
Google

Rose Chemeli Ledama

32 yrs (May 15, 1980)

OpenMRS Identification Number: 2900EG-8

BMI: ? (Weight: 62.0 kg , Height:) CD4: 261.0 | Regimen:

Start Visit
Overview
Regimens
Visits
Demographics
Graphs
Form Entry
Consultations

Select a Doctor

Doctor name: Dr. Jack Sheppard

Problem Description

Description of the problem:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Patient's Records

Record Type	Creation Date	Record Owner	Authorize
Blood Analysis	06/22/2011	Dr. John Watson	☑
Test HIV	02/18/2012	Dr. John Watson	□
Personal Information	04/03/2012	Dr. John Watson	☑
Radiology	12/17/2012	Hospital of S.Francisco	□
EKG	12/17/2012	Hospital of Trento	□
Consultation Response	12/21/2012	Dr. Jack Sheppard	□

Figure 3.18: A screenshot of the consultation request form of the CHINO OpenMRS Module deployed on OpenMRS.

with business process designer to test if they were able to model the CHINO processes according to identified requirements.

3.8.2 Usability Validation

According to the CHINO methodology, Business Analysts and Developers should be able to define the processes in compliance with the identified requirements by using the Modelling Framework. To test these assumptions and the Modelling Framework usability, we performed a user study with a group of nine developers that had preliminary knowledge about process modelling with the standard BPMN Activiti Designer [2]. With the user study we tried to understand if the requirements identified at *Steps 1, 2 and 3* can be mapped into business processes at *Step 4*. The users where chosen among master students and employees of the University of Trento. The analysis was based on notions from the Interaction Design (ID) studied in Human Computer Interaction (HCI) discipline and applying the usability testing methodology called Think Aloud [157].

According to it, the standard usability test is performed recording users performance on an assigned task. In our test we showed to users a document explaining the CHINO framework, an data sharing scenario (e.g. doctor consultation) and a list of identified policies from Italian [94] regulations. We monitored and stimulated them to speak while performing the assigned tasks to analyse their behaviour. At the end of the test we asked them to fill a questionnaire about overall satisfaction about the assigned tasks.

The objectives of this test were:

1. Understand if CHINO Modelling framework is easy to use and identify potential usability limitations.

2. Evaluate if the Custom Tasks are easy to understand and to use and identify possible improvements.

The Study

Prior to the main test we did a pre-test with two university employees to check if the provided information and documentation was clear enough to complete the assigned tasks.

The pre-test gave us few important feedback to improve the main test. For example a user pointed out that in the case of the Custom Task named *Wait Approval*, it was easy to understand its meaning but difficult to understand how to use it and its behaviour. At the same time, the other user raised some questions about the component *Ask for Access Rights* and its usage in conjunction with the task *Send Request Wait Message* because of the ambiguity of the second one as shown in the Template 3.10. To overcome these limitations we updated the documentation about Custom Tasks usage and performed the second test.

During the main test we asked some questions to users that had two types of responses. The first one in a scale from 1 to 7 points where 1 correspond to negative opinion such as *Strongly Disagree* and 7 to a positive judgement such as *Strongly Agree*. The second type was open questions. All the numeric questions were mandatory while the open ones were optional. We report some questions while the complete questionnaire can be found in [81].

Q1 "Overall, I am satisfied with the ease of completing the exercise in this scenario."

Q10 "I was able to complete the exercise quickly using this system."

Q21 "This system has all the functions and capabilities I needed."

Q23 "It was easy to understand the concepts introduced by this framework."

Q25 "How do you rate the overall experience with the CHINO Modelling?"

Results

To evaluate the responses for each question we calculated the mean (μ_n) and variance (σ_n^2). According to our unit of measurement higher values are better, while lower variance is preferable since the first coefficient expresses the positive or negative opinion of the users, while the second represent the level of disagreement among them.

According to our analysis the test showed a positive impression about the tool usage after few times it has been used. However, when users used it for the first time some differences among opinions emerged. Only two users expressed a negative feedback about their performance and usage of the modelling framework. However, since they were able to perform their tasks, this does not represent an important limitation, although it suggests taking into consideration different approach in the training of new user to CHINO for the first time.

Overall, from the test it emerges a good opinion about the modeller usability. Moreover the open questions gave us some positive feedback:

> "I am comfortable with the diagrams because it really represents the information which is held on hospitals."

And also some negative ones:

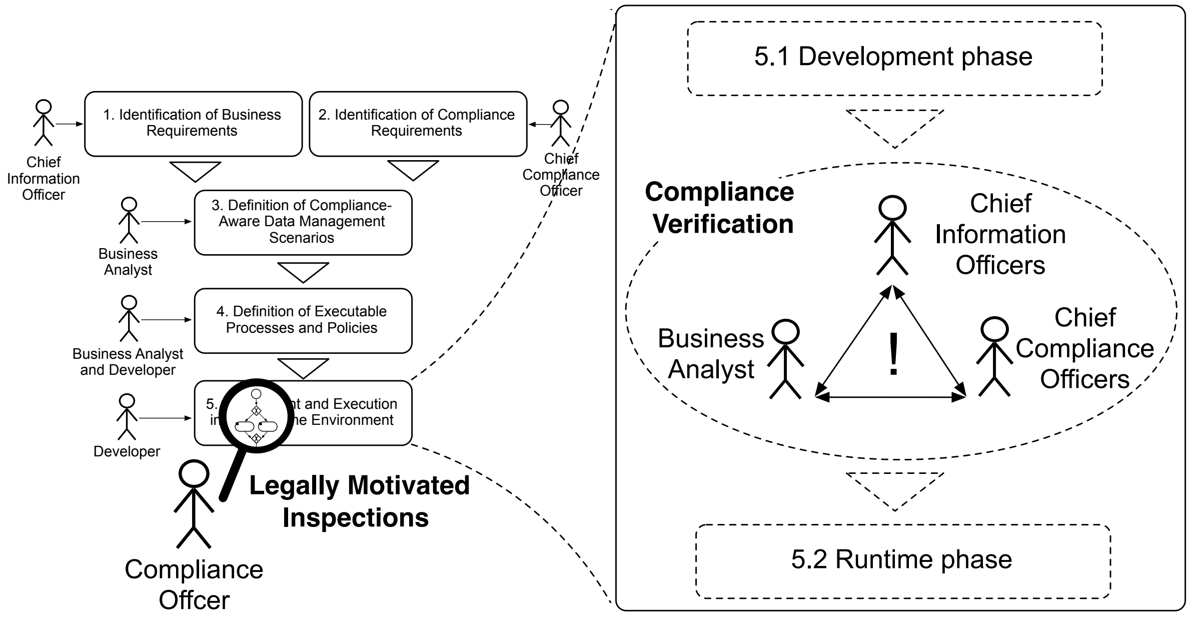
> "The framework as I said is easy to use but anyway I had some problems of stability during the usage, so for this reason, relatively to the question if *I would recommend this tool to others* the real answer is yes, but..."

> "The Activiti designer in general presents some problems like sometimes it freezes."

The stability issues are related to the Activiti Designer and not to our specific extension but it is just a matter of software maturity since Activiti team is releasing frequently new versions of the Designer.

Overall, the User Study gave us important feedback about Custom Tasks usability and suggested some improvements especially regarding the explanation of their usage. Other suggestions include also the need for better explanation of usage of combinations of different tasks to achieve a specific goal. Overall, tests showed a satisfactory usability level of the Modelling Framework and demonstrated that users were able to transpose requirements into processes while underlying the need for smaller improvements of the CHINO platform.

With these tests we validated the technical usability and feasibility of the CHINO approach, while the next section analyses how CHINO components can contribute in achieving privacy law compliance.



Chief Information Officer
1. Identification of Business Requirements
2. Identification of Compliance Requirements
Chief Compliance Officer
Business Analyst
3. Definition of Compliance-Aware Data Management Scenarios
Business Analyst and Developer
4. Definition of Executable Processes and Policies
Developer
5. …t and Execution i… …e Environment
Legally Motivated Inspections
Compliance Offcer
5.1 Development phase
Compliance Verification
Chief Information Officers
Business Analyst
!
Chief Compliance Officers
5.2 Runtime phase

Figure 3.19: CHINO Methodology with the focus on compliance inspections and verifications.

3.8.3 Privacy Law Compliance with CHINO

Here we analyse CHINO from the legal point of view by involving a privacy and compliance expert. We reason about its ability to preserve privacy and data protection rights and to support compliant processes definition. We try to answer in particular to the following two macro-questions:

1. If CHINO provides all the technological elements (modeller, modelling elements, internal components) to support the development of privacy law compliant data management processes and policies.

2. If CHINO could facilitate the tasks (emphasised in Figure 3.19) of process and policy approvals or verifications that is done before going into production phase, and the legally motivated inspections by Compliance Officers at runtime phase.

In order to answer to the first question we start by analysing the recommendations of the Art. 29 Data Protection Working Party in [18, 15]. Working Party provides recommendations on several topics emphasising the need for special safeguards in order to guarantee the data protection rights of patients and individuals. Some of recommendations include the respect for data subjects' self-determination and authorisation procedures, security measures, transparency, liability issues and finally, the availability of mechanisms to control the data processing.

As described in the chapter, CHINO aims at providing an effective framework to support the privacy by design approach that has been identified as one of main principles in the development of systems that manage privacy sensitive data by the Art. 29 Data Protection Working Party in the document The Future of Privacy [16] and by the European Commission's proposal for a new Data Protection Regulation [72]. Moreover, CHINO proposes a proactive approach based on data self-determination in accordance to the privacy by design principles, providing effective technical and organisational tools for healthcare institutions.

Analysing more deeply CHINO with the focus on data protection requirements, it appears to be an appropriate platform for sharing personal and healthcare data also among organizations that belong to different regulatory contexts [70]. From the data security point of view, CHINO provides the necessary mechanisms to satisfy the security requirements related to healthcare data management according to Art. 31 and 33ss of the Italian Data Protection Code [83] and to the Privacy Impact Assessment [71] of the European Data Protection Regulation [72]. It implements technical and organisational features to avoid loss or unauthorised alteration, processing and access to data. Furthermore it respects data protection general principles from the Directive 95/46/EC [69], and in particular the principles of purpose limitation, proportionality, data quality, necessity and the data subject's rights.

CHINO is able to enforce the explicit consent policy that is defined as the data subjects' explicit consent on the processing of their data and it is an exemption to the general prohibition to personal data processing, according to European legislation (Art. 8, Directive 95/46/EC) [69, 17]. CHINO access right policies and the assurance mechanism enable data subjects to freely express explicit, specific and informed consent about data sharing. According to the legislation, in special cases data can be processed without consent (e.g. compliance with legal obligations, protect vital interest of data subject, public interests). This is possible in CHINO by defining special conditions on the Check Access Right modelling element. Processes can be also defined to delegate the disclosure of data to data subjects' personal doctors. Data subjects could also delete and block data sharing according to the Italian legislation (see Art. 7, Italian Data Protection Code [83]). Moreover the involved actors are able to receive notifications about the process status, including the requests of access. The updates of wrong data to assure data quality policy according to Italian, European and HIPAA legislations, are done through the Push Record task.

According to European legislation (Art. 6 of Directive 95/46/EC [69]) and to the Art. 11 of Italian Data Protection Code [83], personal data can only be processed for specified explicit and legitimate purposes and may not be processed further in a way incompatible with those purposes. CHINO provides technical tools for enabling data controllers to check step-by-step the lawfulness of the personal data process following the purpose principle [19]; the legitimate purposes of the process are recorded and all the access requests are filtered according to them.

CHINO provides mechanisms to release data only according to the specified, explicit and legitimate purposes through the definition of filtering policies. Namely, the CHINO filtering task provides anonymisation mechanisms to remove sensible information on a purpose-based approach. For example in case the data need to be used for statistical purposes, a filtering policy that eliminate personal identifiable information can be defined [11, 164].

By analysing more deeply the data security features, CHINO guarantees confidentiality and integrity of information against unauthorised access, disclosure or alterations. Moreover, it improves personal data traceability, so that each communication and each data transaction can be tracked back to a certain entity that can be easily audited. In order to assure data traceability, CHINO provides features to clearly identify all the actors and entities involved in the process execution. This allows identifying data controllers and data processors (and other involved entities) when executing operations over data and addressing specific and defined liabilities to data controllers and processors at any step of the processing. Logging ensures accountability on operations over data in compliance with Articles 28ff of the Italian Data Protection Code [83] and with the Guidelines on the EHR development [94]. CHINO allows data controllers to keep privacy-sensitive data on their own servers if they have restrictions about data storage administrative locations, as it is the case in Italy [94]. Regarding the data stored inside CHINO, it is encrypted with standards algorithms (e.g. AES-128 and SHA-258 for hashing). The deployment of CHINO could be done also in Cloud-based environments. Although this aspect needs a deeper analysis, the combination of the possibility to decentralise record storage and encryption techniques satisfy the requirements imposed by Art. 29 Working Party [15].

Relatively to the second question, we tried to analyse the healthcare software lifecycle that is depicted in Fig. 5 with particular focus on the compliance aspects that have underlined by two specific situations. Namely, the Figure 3.19 shows the situations where the "Chief Compliance Officer", that is usually a privacy expert or a Data Protection Officer, is involved in the verification of the business processes developed at Step 5 and has the responsibility to approve or reject them. The other situation is related to recent Inspection Plan [84] undertaken by the Italian Data Protection Authority in which medical record systems has been included as one of the potentially analysed systems. This means that the Data Protection Authority will seek for documentation to check if the data lifecycle and data management procedures are compliant with legislation in order to assure protection to data subjects' rights. Both situations describe tasks that could have significant potential impact on projects developed without considering exhaustively privacy related aspects (i.e. fines to responsible organizations or, in extreme cases, systems suspension or disposal).

In such direction, CHINO is able to facilitate inspection procedures due to its adoption of BPM technology to define data management operations. Similarly to other scenarios and context [29, 147, 152], visual representations in CHINO simplifies the process of revision by lawyer and privacy experts due to its simplification of understanding for people with non IT background. CHINO expresses in a more clear way which privacy requirements are satisfied when compared to standard textual documentation making easier to identify different steps and related rights, duties and liabilities.

3.9 Lessons Learned and Discussion

Regulatory compliance is a complex goal for every organization that deals with sensitive data. In healthcare, this is even more difficult since regulations and practices are complex and vary from country to country as well as over time. The two regulatory contexts (Italy and UK) that we analysed and the data exchange scenarios we identified motivate the need for a system that is able to manage different privacy and security rules and orchestrate policy enforcement points and data stores across different healthcare organizations that can potentially belong to different regimes. To help organizations in identifying, defining and executing such regulatory and organization specific requirements, we proposed a new methodology and an execution environment to:

- Capture the sequence of steps that need to be carried out by organizations to define their own security and privacy policies and data sharing processes to conform to high-level regulatory compliance requirements.

- Identify a set of elements, IT components and actors that can be orchestrated by data sharing processes to achieve compliant data sharing.

- Provide an environment for the definition and execution of shared processes and policies to support data, process and policy management.

The overall approach is based on a novel use and customization of business processes to model the internal business logic of data management operations that allows involved actors to manage and share data while achieving compliance. Processes serve as a vehicle to achieve high customization of operations in such a way that each involved actor can satisfy its privacy, security and business requirements. In particular, CHINO is able to execute data owners' processes and policies when their data is accessed by other organizations. In such way, while the organizations are using the same set of interfaces to interact with CHINO, they execute policies and processes of data owners and thus achieve a compliance-aware data sharing. It is important to clarify once again that we tackle only technological aspects while data semantic is out of scope of this work.

Furthermore, the business process based approach allows us to better understand and share our understanding of compliance requirements, and to reason about the process definition and process improvements. The visual modelling of business process allows us to achieve better visibility and transparency on security and privacy rules. This can help at improving the trust and compliance among the participant healthcare organizations in terms of data sharing.

From our experience, we observed that defining methods for sharing privacy sensitive data is a multidisciplinary task that involves business, IT and privacy experts. The process-driven approach could provide a vehicle for the involved actors to design, develop, verify and improve more easily compliance policies.

CHINO can be seen as a general-purpose privacy-aware data management platform that can be used also in other domains that are characterized by similar data sharing and accessibility requirements.

Chapter 4

Architectures, Protocols and Policies for Privacy-Aware Medical Record Sharing

This chapter describes how we tackled the challenges related to service integration in a socio-health scenario and how we developed and validated a privacy-aware data sharing protocol, underlying privacy policies and the supporting technology. These contributions have been incorporated within the previous chapter to support process execution and development of the CHINO technology.

We start by showing how the lack of interoperability among agencies delivering care makes the services (i) inefficient for the service providers, (ii) difficult to access for the citizens who need to bring along paper-based records and (iii) hard to monitor and assess for the bodies in charge of the governance.

We then describe the analysis approach to identify the data dependencies among institutions (i.e. the data they produce, consume and need). To support data sharing we propose a privacy-aware event-driven protocol and a system prototype. In the considered integration scenario the protocol delivers the data to an Electronic Health and Social Record and a Business Intelligence system. The proposed privacy policies provide a tight and incremental control on the access and dissemination of sensitive information.

Before concluding, we report the validation phases including an on-field experimentation in which an instance of the developed system has been deployed to integrate the social and health services in the Trentino Province (Italy).

4.1 Introduction

In the last 30 years we have witnessed a considerable increase of life expectancy worldwide [182] and consequently an increase of elderly population that, in many developed countries, is almost exceeding the number of people in working age, with serious economic and social consequences. In Italy, as it is the case in rest of the Europe, large families are rare and single-person families are becoming common, including families composed of an elderly person alone. A common strategy adopted to reduce the public expenses to manage this phenomenon consists in facilitating the autonomous and assisted life of elderly at their houses [160]. Long term care of elderly (and, more in general, of fragile people of any age with physical and cognitive disabilities) at home requires a breadth of services from sanitary and social domains.

In general the processes for applying and obtaining such services are fairly complex, since they involve multiple private and public institutions that receive and process the requests, evaluate the applications and determine priorities and service levels, deliver the services, assess the quality and cost of services delivered. The result is a complex cross-organisational process to be executed each time a new service request arrives.

In this chapter we study the problem of privacy-aware integration of social and health services and we show the solution we developed and how it has been applied to a set of use cases in Trento (Italy). The use cases are extracted from a project undertaken by the Autonomous Province of Trento that involves a dozen of institutions from the IT sector, the public administration and the healthcare services.

We provide a solution to governmental bodies which are interested in facing two categories of needs: *Business Intelligence* and *partial automation of assistance services*. Governmental bodies need Business Intelligence on the quantity and quality of the services delivered to citizens, both to ensure that proper assistance is provided and to establish the amount of reimbursement due to the agencies providing the services. In absence of integration solutions, these indicators are collected manually, sporadically, and with different practices at each institution delivering services. In such cases governance spends considerable amount of time to compute the indicators and the obtained results are often unreliable.

The second need is the partial automation of the cross-organisational assistance processes. While the first objective is of interest mainly to the governance, the second is of interest to citizens and institutions delivering care, as they aim at providing more efficient and reliable assistance services.

From a technical and organisational perspective, the development of solutions for this kind of problem is very challenging:

1. It departs from “traditional” data integration by requiring a process integration, with the added complexities of being cross-organisational and characterised by a large number of medium and small institutions that also dynamically grow over time (civic centres, hospitals and social care institutions will need to progressively join the initiative);

2. Strict privacy rules defined by data protection legislations [83], guidelines for health

records management [94] and proposed architectures at national level [143] forbid the adoption of traditional data warehousing and integration approaches. Those rules forbid collecting data in a central repository and define strict legal constraints on the way data is collected, stored and distributed in a context with multiple organisations. Such constraints make it difficult to identify application protocols and policies for such kind of integration. Since the Italian Data Protection laws and Health Information Systems trends are similar to many EU [69, 136] and non-EU [36, 76] countries, the problem is quite general.

In this chapter we report and discuss how we addressed and solved the identified challenges and how we validated the developed solution on a set of real case studies. The following are the main contributions of this chapter:

1. It shows how a data integration problem in a multi-organization and rapidly evolving environment can be addressed via a process- and event-based approach which makes it easy for new institutions to come on-board, minimises the development and maintenance effort required for the integration, and - perhaps most importantly - blurs the distinction between a data and a service integration, providing institutions with the benefits of both;

2. It describes how privacy and data sharing can coexist thanks to a protocol that meets regulatory requirements via privacy policies defined by data sources. It restricts the access to information only on-demand and supports the data sources in the definition of fine-grained privacy policies constraining who can see what and for which purpose;

3. It presents the architecture and implementation of a solution that achieves integration of health and welfare services that has been deployed in Trento, and discusses the many lessons we have learned in doing this.

An instance of the proposed solution is now being rolled out in production after successful experimentation with the involved institutions and the source code has been also released under the GPL v3 free software license [77] and made available at the European Commission's Joinup repository [48].

In the following we begin by describing works done in related areas. In Section 4.2 we describe the project context and goals. Section 4.3 describes the analysis approach we followed, the system architecture and its main components. Section 4.4 details on privacy aspects while Section 4.5 shows the testing of developed system with involved stakeholders. We discuss the observed benefits of our approach and we conclude in Section 4.6.

4.2 Motivating Scenario

In the Province of Trento, as in the rest of Italy and in many countries worldwide, the welfare agency delivers a set of public services through many smaller agencies and municipalities that do not share the same information systems; instead they have their own, sometimes self-developed, custom software applications to satisfy their needs. To be reimbursed and to fulfil agreement requirements imposed by the public sector, they need to send periodically accounting and statistical information to the central welfare agency. For the welfare agency, collecting such information, carrying out refunds and obtaining visibility on the quality and the economy of service delivery, requires collecting and integrating information from literally dozens of completely heterogeneous and fairly complex systems. Furthermore, the lack of coordination among agencies and the complexity of assistance processes badly affect the quality of the services delivered as perceived by the patients with delays and lack of visibility on the progress of their requests. In addition, the organisations waste time transferring information from paper-based documents into their information systems with possibility of errors. Due to these inefficiencies, the Province started a project that aims at addressing two main categories of needs:

1. The enabling of the organisations involved in social and health assistance to exchange information among their information systems and to create an Electronic Health and Social Record (EHSR) for patients.

2. The automation of information gathering about Key Performance Indicators (KPI) and cost metrics that are defined by the governance for the social and health services. A list of these KPIs and metrics is provided in Table 4.1;

Figure 4.1 depicts the project scenario and the role of the infrastructure that we designed and developed to satisfy project objectives.

The actors involved in the project scenario and their responsibilities can be classified using the following categories according to the privacy regulations [94]:

- Data Subjects: citizens and patients to whom the data relates;

- Data Controllers: the Healthcare agency and socio-health service providers, the local municipalities and groups of districts, private companies and organisations delivering tele-assistance services, nursing home services and long term assistance in elderly houses or recreation centres. They act as data producers and consumers as depicted in Fig 1 while the Governance acts only as a data consumer;

- Data Processor: the developed system that acts as a data-sharing platform.

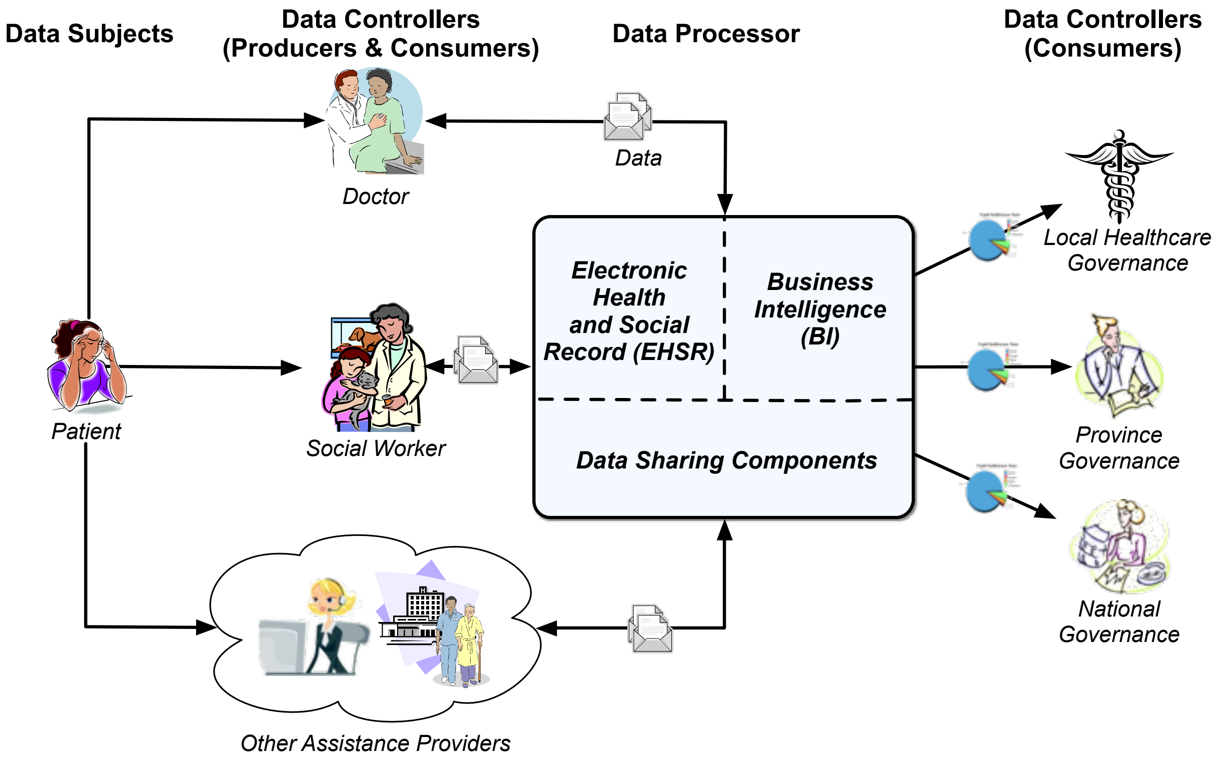

Data Subjects
Data Controllers (Producers & Consumers)
Data Processor
Data Controllers (Consumers)
Data
Doctor
Patient
Social Worker
Electronic Health and Social Record (EHSR)
Business Intelligence (BI)
Data Sharing Components
Local Healthcare Governance
Province Governance
National Governance
Other Assistance Providers

Figure 4.1: The role of the developed system including the data sharing components, the Electronic Health and Social Record (EHSR) and Business Intelligence (BI) services.

Due to the complexity of the scenario, the project considered only a subset of four assistance processes while the entire set of health and social services is more than a hundred subdivided in different areas [144]. The selected services are listed below:

1. Assistive and healthcare services for elderly with disabilities delivered directly at home with the help of nurses, family doctors, social workers and private cooperatives delivering meals and house cleaning services;

2. Long term healthcare services in specialised structures (e.g. rest homes);

3. Recreation centres for elderly people providing transportation, daily activities and meals supply;

4. Tele-assistance services with a 24h call centre checking periodically the state of the assisted person and reacting to critical problems (e.g. emergency calls from the user).

Given such incremental approach, one of the main identified requirements that a system needs to tackle is facilitating the joining phase of other institutions to the initial group after the

Involved Organisations	Requested KPIs
Province of Trento: collects information on the services delivered to monitor their quality, resources, reimbursements and budget planning. Demographic evolution per age classes;	# of assistance requests per district.
Welfare agency: evaluates cognitive and social state of the patients to complete the requests of activation of socio-assistive services.	# of patient per social workers; % of accepted requests; # of requests per territory area.
Healthcare agency: evaluates health state of the patients to complete the requests of activation of socio-assistive services.	# of requests of assistance by requestors (general practitioners, hospital doctors, social workers)
Local municipalities and districts: manage the administrative procedures and activation of the socio-assistive services, financial support and delegation of service provisioning to accredited organisations.	Average cost of services per person; # of administrative practices completed within 60 days.
Private companies and no-profit organisations (e.g. tele-assistance), nursing home services, long term assistance in elderly houses, recreation centres: deliver the final services to the patients and interact with their family doctors, relatives.	# of alarms per type (healthcare/ social alarm, monitoring device failures); # of hours of services per patient.

Table 4.1: A subset of KPIs the organisations are interested in.

first pilot, since not all participant organizations were involved. This is particularly challenging as the systems used in each organisation are very heterogeneous with solutions implemented both in-house by dedicated IT departments or acquired by third party IT companies.

Solving this heterogeneity by adopting one single solution would have the advantage of improving the homogeneity in procedures and nomenclatures as well as facilitating communication. However, it would introduce the drawbacks of imposing a generic system that cannot reflect all the specificities of single districts and organizations, and it would disrupt the current political and organizational configuration. For these reasons, there is the need for a conservative approach to preserve the systems currently in use.

Regarding the collection of KPIs, Table 4.1 gives some examples used for budget planning and monitoring of the service quality. They are also used to monitor the service provisioning and its compliance with the quality of service constraints defined by the Province (e.g. the elapsed time from the approval of tele-assistance service request until its activation cannot be longer than 7 days).

Collecting such KPIs is challenging since they are intimately process-related, although the underlying business process is very often poorly stated and defined. In fact, the indicators relate to specific events of single assistance processes - i.e. they can be calculated by tracking what happened and by discovering relationships amongst the events. To discover such events, it is not reasonable and feasible to fully analyse the assistance processes: they have an intrinsic and necessarily “unstructured” behaviour, owing to the managed services, which depend on human decisions and evaluations. In these cases, the traditional data warehouse approach does not work; it is already lengthy and hard to integrate a few systems in a single organisation, let alone integrate dozens of them that are developed and managed by different institutions and that execute unstructured assistance processes.

Next section describes how we approached the development of the system to tackle this scenario starting from the analysis phase and then describes the data that are exchanged and the event-based architecture we developed to tackle the specific project requirements.

4.3 Event-driven Architecture for Privacy-Aware Data Sharing

The case study described above is representative of a class of problems requiring data and process integration while preserving privacy. A common approach in data integration starts, naturally, from the data itself. However, our experience in the healthcare domain is that this is hard to do for several reasons. Namely, the challenges are both technical and organisational and are related to the domain complexity of social and health services and heterogeneity of the involved institutions:

1. The service providers have heterogeneous IT systems having large and complex databases;
2. The IT departments of the involved institutions are sometimes not adequately staffed;
3. The integration is related to the processes that generate the data, not only to the data itself.

For these reasons, the methodological approach we follow is to tackle integration by analysing and focusing only on assistance processes of interest and by identifying, with the relevant stakeholders, the points in the process at which they need to inform other institutions about an event (and related information) of interest that occurred during the process.

To identify processes and events of interest, one of the challenges consists in keeping the stakeholders focused in a short-term joint effort as they may have different expectations and backgrounds ranging from sociology, medicine, and financial management to IT. Their involvement is crucial for projects success since the assistance processes are not documented and most of the times they are only in the people’s mind. For these reasons, before showing how we tackle the technological challenges and the technology we developed, we start by describing the analysis approach we propose.

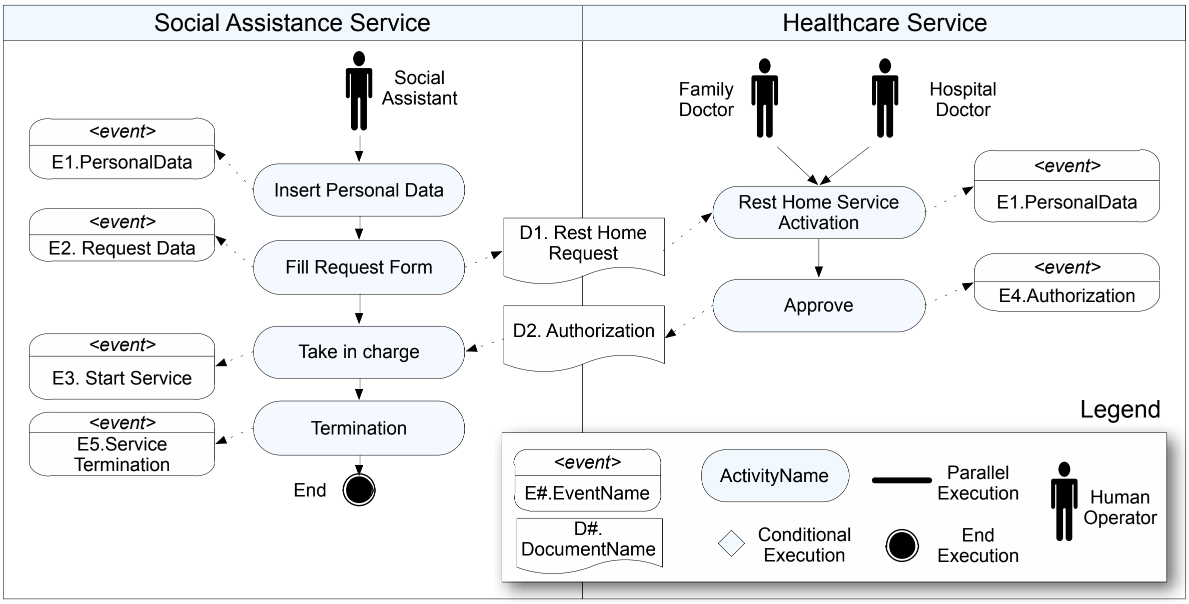


Figure 4.2: An example of activity diagram with focus on events generated and documents exchanged among service providers.

4.3.1 Process-Oriented Analysis

The analysis approach we propose focuses on the workers-application interaction in order to capture the data produced at each step with the twofold goal of:

1. Isolating the points of cooperation and interoperability among the inter-related portions of assistance processes executed by the parties - i.e. developing a common domain of accepted concepts, shared amongst the organizations and the actors;

2. Identifying data of interest that could be used for feeding the Electronic Health and Social Record (EHSR) or the Business Intelligence (BI) modules. While the BI needs to collect data to provide a comprehensive analysis of the business processes occurring among the parties, the EHSR needs to collect data for building the patients' socio-health medical history.

The approach is inspired by collaborative analysis and task oriented analysis techniques [127, 173] that as opposed to other methodologies such as ethnographic methodology [32] which requires considerable amount of time to complete the analysis, allowed us to gather a concise representation of the integration scenario by involving the domain experts. The outcome can be given to IT designers and analysts to gather the set of requirements, prioritise and rapidly translate them into system specifications.

To facilitate the analysis we represent the assistance processes and generated events graphically by adopting activity diagrams. A simplified example is shown in Figure 4.2 while an example from the case study is shown in Section 4.5.

We chose activity diagrams for their simplicity and understandability [62, 153] but also other modelling formalisms could be used such as Business Process Model and Notation (BPMN) [139]. We add to the standard notation some stereotypes to represent the data (both on paper and in electronic forms) that are produced during each activity. In Figure 4.2 rounded white boxes represent information generated inside IT systems (E1, E2, E3, E4, E5) while squared white boxes (D1 and D2) represent documents exchanged between the two organisations.

With this approach we identified the information of interest for the interoperability and the BI in form of *events*. Intuitively, an event is the occurrence of a change in a system and it contains contextual information like the author, the data subject, the reason (i.e. the type of the event), timestamp, and a payload representing what happened (e.g. the Rest Home Request). In such way we could omit further analysis of sources' information systems such as database schemas or the technology. This approach instead allowed us to deal with assistance processes which are not well defined and documented and may have many exceptions. For example they may start from a social worker but also from the medical staff and each sub-process can end at any point in time (e.g. for rejection of the request or death).

In Section 4.5 we report a detailed description of the steps we followed in our case study and we show an example of a diagram for one of the analysed assistance scenarios. We show how the process-oriented analysis supported by visual representations involved the actors during the group meetings making them proactive during the definition of activity diagrams and the content of each event.

Next subsections show the data sharing protocol and the system architecture that we designed and implemented to share the identified events.

4.3.2 Data and the Data Sharing Protocol

The identified set of legal [83, 94], technological [143] and organisational requirements that need to be satisfied relatively to the class of problem we aim at solving, as demonstrated by the case study, are:

1. The minimisation of the commitment of the institutions to join the infrastructure facilitating the exchange of information and minimising at the same time the traffic (that is, ensure that institutions get information on an as need basis);

2. Avoid duplication of sensitive data outside the boundaries of the data controllers;

3. Ensure that sensitive information is delivered only upon data producers provided authorizations for stated purpose of use;

```
<Metadata>
      <eventId>1231321</eventId>
      <senderId>345</senderId>
      <senderDesc>Soc. Ass. Service</senderDesc>
      <receiverId>5463</receiverId>
      <receiverDesc>Health. Service</receiverDesc>
      <name>Mario</name>
      <surname>Rossi</surname>
      <birthDate>1918-07-23</birthDate>
      <SSN>MRIRSS18L23233Z</SSN>
      <timestamp>2014-10-30 T 11:25</timestamp>
      <recordType>55</recordType>
      <recordDesc>RestHomeReq</recordDesc>
</Metadata>
```

Figure 4.3: An example of metadata message for the Rest Home Request Service.

4. Inform involved parties about information availability that is potentially of their interest (without disclosing confidential information), so that they can then explicitly ask for the confidential information by specifying also the purpose of use.

To tackle these objectives, we designed two kinds of data (inspired by EHR guidelines [143] and international standards such as IHE - Cross-Enterprise Document Sharing (XDS) profile concepts [93]) that are characterised by different levels of sensitiveness:

- *Metadata* describes a record and is used to signal that some event has occurred in a legacy system. *Metadata* contains only information on the context in which the event occurred such as: the data subject (patient/citizen); what happened (type of event, assistance service); when; who generated that information (the data producer organisation and its system), and potentially to whom should be delivered;

- *Record* contains all the data to fully characterise the event that, by default, should be kept secret and shared only upon explicit authorisation of the data producer. It contains detailed and privacy sensitive information (e.g. the Rest Home Request shown in Figure 4.4). A *record* can contain any type of content (healthcare, administrative, financial).

Examples of *metadata* and *record* (simplified for readability reasons) that are generated at step "Fill Request Form" in Figure 4.2 (corresponding to event E2) and are included (i.e. printed) into the document D1 are shown in Figure 4.4.

```
<Record>
      <recordId>23</recordId>
      <recordTimestamp>2014-10-30 T 11:25</recordTimestamp>
      <recordType>55</recordType>
      <recordDesc>RestHomeReq</recordDesc>
      <content>
            <socialStatus>Autonomous</socialStatus>
            <economCoeff>2.3</economCoeff>
            <partcicipCoeff>0.6</participCoeff>
            <probDesc>The requester is autonomous
                  from cognitive  and psychological point
                  of view. The physical state is normal.
            </probDesc>
            ......
            ......
      </content>
</Record>
```

Figure 4.4: An example of record message for the Rest Home Request Service.

The establishment of a common understanding of terminologies among involved organisations is key to interoperable data sharing [93]. For this reason we define the structure and semantic meaning of the *metadata* and the structure and envelope of the *record* message. However, to enable the record message to encapsulate and transport any type of data, we do not define its internal structure and therefore also the semantic meaning. Although HL7 [60] represents the de-facto standard for encoding and exchanging healthcare data, other types of content (e.g. administrative and financial) need to be encoded in different formats. Therefore, as shown in Figure 4.3 and 4.4, we adopt XML as the encoding language of *records* and we define the tag <content> of type *any object* to deal with different contents varying from message to message. This approach gives the possibility to organisations to exchange any XML object without changing the record retrieval operation interface (API). The tag <recordType> allows the data consumers and the data processor to know the type of exchanged *record* and parse its content (e.g. to apply privacy policies and filter unnecessary data for specific data requests). An XML schema [176] is associated to each of the record content types and it is shared among the parties. This approach leaves to the involved organisations, or to a responsible governmental body, the task to address semantic interoperability at the information model level by defining the record content structures and semantic meaning. In case other data representations will be

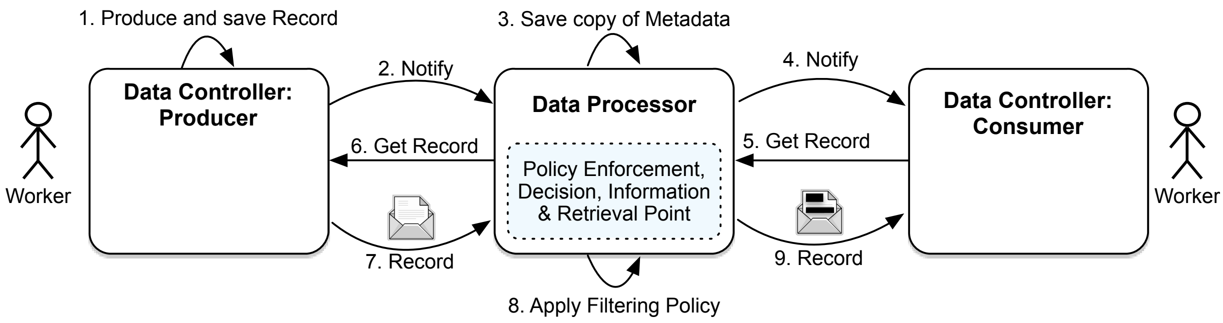

1. Produce and save Record
3. Save copy of Metadata
2. Notify
4. Notify
Data Controller: Producer
Data Processor
Data Controller: Consumer
Worker
6. Get Record
5. Get Record
Policy Enforcement, Decision, Information & Retrieval Point
Worker
7. Record
9. Record
8. Apply Filtering Policy

Figure 4.5: The sequence of steps to achieve data sharing among data producers and consumers.

introduced, the only modification will be needed to the Visibility Rule Manager (detailed later), which applies filtering policies based on the content of records.

Data-sharing Protocol

To exchange *metadata* and *records* we extend the concepts proposed by the IHE - Cross-Enterprise Document Sharing (XDS) profile [93] by introducing privacy-awareness given by the *filtering policies* and by making it event-driven. Although the XDS protocol has been proved as effective to exchange medical records in many projects [143, 36], it lacks a fine-grained filtering of data based on the purpose of use (e.g. BI or healthcare assistance). Namely, in contexts characterised by a high heterogeneity of data, as it is in our case study, delivering the appropriate set of data for specific purposes should be faced at infrastructure level (i.e. before delivering data to the BI module). Therefore we define a protocol that adds filtering of data at sub-document level, in addition to a two-phase incremental sharing of *metadata* and *records*, to achieve privacy-aware data sharing. The overall sequence of operations that data producers and consumers need to perform in order to establish data sharing is described by the sequence of steps shown in Figure 4.5.

After the producer has generated the *record* (i.e. an event in Figure 4.2), it saves it on its internal record store (Step 1 in Figure 4.5) and sends related *metadata* by notifying the central data processor (Step 2). The data processor delivers the *metadata* to all the interested consumers that have subscribed to receive that specific event type via a publish/subscribe mechanism managed by the *Service Bus* (detailed in the next subsection). If the receiver field has been specified, then the message is sent just to that specific consumer and, in case it is of statistical interest, to the BI module. The Service Bus forwards also a copy of the message to the EHSR that is subscribed to all events by default. This approach decouples producers and consumers enabling the later to subscribe to events of interest based on their needs.

When the data consumer receives *metadata*, it can ask for record content calling the *Ge-*

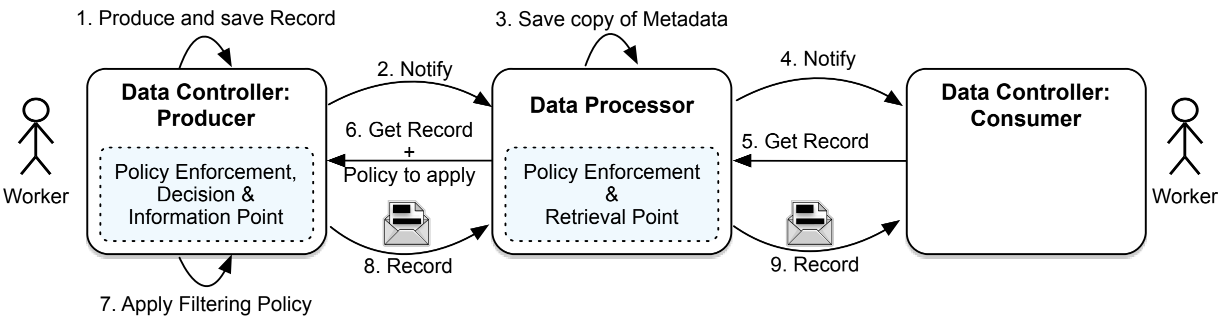

1. Produce and save Record
3. Save copy of Metadata
2. Notify
4. Notify
Data Controller: Producer
Policy Enforcement, Decision & Information Point
Data Processor
Policy Enforcement & Retrieval Point
Data Controller: Consumer
6. Get Record + Policy to apply
5. Get Record
Worker
Worker
8. Record
9. Record
7. Apply Filtering Policy

Figure 4.6: The sequence of steps to achieve data sharing among data producers and consumers by applying policies at data producer side.

tRecord web service operation. To fulfil the request, the consumer will be asked to specify the purpose of use of data. The *record* will be restored from the producer record store and it will be filtered by applying filtering polices according to the *purpose* of use declared by the data consumer. To apply such policies, the data processor implements an XACML compliant Policy Enforcement Point [124] (detailed later). Finally the data will be forwarded to the consumer through the data processor.

Although the proposed approach has been successfully tested and validated, the filtering of *records* can be slightly modified to achieve even a higher level of computation distribution and end-to-end security if needed. Such modification to achieve a *decentralised enforcement* is shown in Figure 4.6.

In Figure 4.6 the data processor forwards the *record* request at Step 6 along with the filtering policy that needs to be applied. The data producer applies by itself the filtering policy (Step 7) and returns the filtered *record* to the processor at Step 8, which forwards it to the consumer. This solution improves the privacy and security since the processor does not need to access to the *record* content. Namely, producers could add additional security to the data sharing with consumers by for example encrypting *records* with public keys of consumers [75]. It also decentralises part of the computation to producers' data centres, which applies filtering policies. This scenario becomes particularly relevant to define responsibilities about managing access to data but it adds technological complexity as a drawback. This is due to the need of decoupling the XACML components that applies policies (Enforcement, Decision and Information Points) from the one that manage policy persistence (Retrieval Point) [124]. Although this solution increases the level of privacy and security, for practical reasons related with the project management, the system prototype has been developed and tested according to the scenario in Figure 4.5.

Overall, the data sharing protocol (described by both approaches in Figure 4.5 and Figure

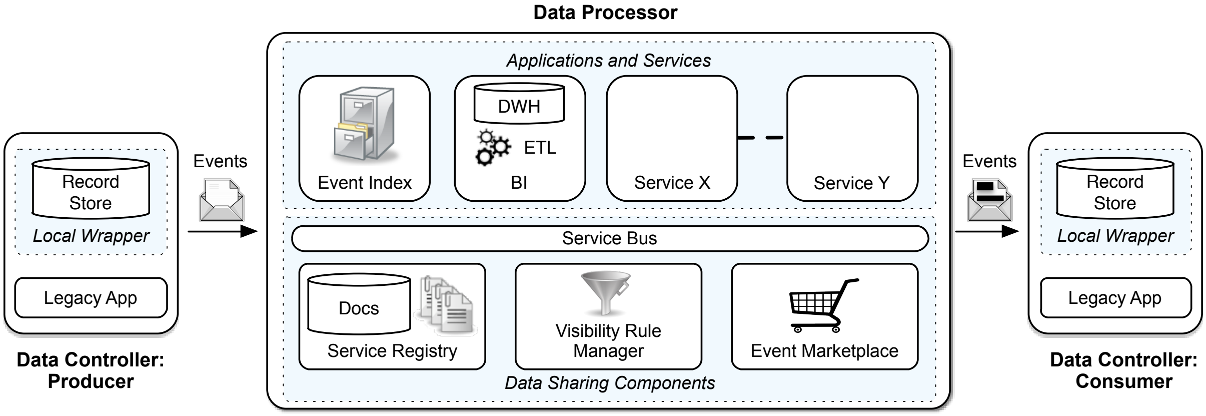


Figure 4.7: The system architecture composed by Data Sharing Components, Applications and Services and Local Wrappers residing on data controllers data centres.

4.6), limits the disclosure of privacy sensitive information and the amount of transferred data since *metadata* contains only a brief description of the *records* and the *records* are kept inside the data producer data centres by avoiding data duplication outside data producers administrative boundaries and in compliance with data protection laws [83, 94] and guidelines for EHR development [143].

To enforce the data-sharing protocol and respect all the other organisational and technological requirements, we designed and developed a system prototype.

4.3.3 The System Architecture

The previously described protocol is enforced by a *distributed architecture* based on *event-driven* and *service-oriented* architectural patterns in which sensitive data (i.e. *records*) is maintained at the sources while the central data processor stores only data references (i.e. *metadata*). In such scenario the data processor mediates the communication among all the parties and acts as a bridge for the routing and distribution of the data and requests. The system architecture that has been designed to support such protocol is sketched in Figure 4.7.

The set of *Data Sharing Components* provides the basic functionalities for data publishing, discovering and sharing. The *Electronic Health and Social Record* and *Business Intelligence* modules are considered as specific “vertical” services laying on the Data Sharing Components and consuming data. To facilitate the joining phase, the platform provides data controllers with a *Local Wrapper*, which stores sensitive data and allows easy retrieval upon requests.

Data Sharing Components

Service Registry stores the contracts, Service Level Agreements (SLA) and other formal and legal documentation that organisations need to sign when joining the platform. Similarly to other projects that aim at interconnecting public administrations [58], the registry is a plain repository of documentation.

Event Marketplace provides a list of all available event types and their description files such as XML schemas [176] and documentation describing the semantic aspects of the *records* content. Participant organisations, based on other organisations' or BI needs, publish on the Event Marketplace the list of events they are able to share. Organisations can explore the Event Marketplace and subscribe to the events of interest. A prototype demonstration of how the *Event Marketplace* operates can be found here [11, 12].

Visibility Rule Manager stores the filtering policies and implements the Policy Enforcement Point (PEP) [124], which applies them on record requests. Data producers define filtering policies by using the *Visibility Rule Manager* (detailed later in Section 4.4), which allow them to control the data disclosure to the specific consumers and purpose of access to data. The purpose taxonomy for the health domain is well defined at national level [83] and purpose-based access control has been preferred to role-based access control mechanisms due to role explosion issues in multi organisation settings [135]. The PEP is implemented by using XACML [124] policy specification language in which obligations are purpose of access to data. The XACML engine is an instance of an open-source implementation of the XACML 2.0 version [66]. In the scenario shown in Figure 4.6, part of the PEP has been placed also inside the Local Wrappers under the responsibility of data controllers. This version decouples part of the Policy Enforcement, Decision and Information Points that are deployed on the data controller side, from the Policy Retrieval Point that is deployed on data processor side [124]. Data processor still acts partially as the Enforcement Point since it receives all the requests that are transferred to the producers.

Service Bus is an instance of the ServiceMix Enterprise Service Bus [8] and implements the operations (APIs) toward the external applications to publish metadata and request records. It acts as the glue component by orchestrating the interaction with other internal components and vertical services that are subscribed to events. Since ServiceMix does not offer reliable message persistence by default, we developed an extension, which provides a persistence module that saves temporarily the data in cases when destinations are unreachable. Afterwards, it tries repeatedly to re-send data using the exponential backoff algorithm [92] until it succeeds. This improves the robustness and reliability of the system.

The list of APIs exposed to external organisations (including the Local Wrapper) is shown in Table 4.2.

The *Notify* API is implemented according to WS-Notification standard [138] that encapsulates messages in a specific SOAP header [177]. WS-Notification is supported by default by ServiceMix and provides a publish/subscribe mechanism that is more interoperable than other event-driven standards such as Java Messaging Service (JMS), which is tied with Java program-

API	Description	Input	Output
Notify	Sends (publish) metadata on the registry and returns the generated unique $metadata_{id}$	$metadata$	$metadata_{id}$
GetRecord	Returns the requested record upon applying the filtering policy according to the specified purpose of use.	$record_{id}$, $requster_{id}$, *purpose*	$record$
DeleteMetadata	Deletes the metadata corresponding to the specified $metadata_{id}$ from the Event Index. The deletion is logical (and not physical) and the metadata status can be restored back.	$metadata_{id}$	$status_{code}$
SearchMetadata	Returns metadata that matches the searching parameters and that the requester is authorised to access.	search parameters	list of $metadata$

Table 4.2: The list of APIs offered by the central data processor.

ming language. The security aspects are tackled at communication channel level by adopting RSA public-key encryption schema [75]. All the activities performed by the data processor are logged to support audit activities.

Local Wrapper

Medical data requires very long retention period, as it should be granted accessibility for the whole patients' lifetime plus a fixed number of years depending on regulations [83]. To facilitate data controllers on achieving such requirements and facilitating the connection with the central platform, we provided a wrapper module with a local record repository. Besides helping them in storing a copy of the records, the Local Wrapper:

- Facilitates data controllers in joining the infrastructure as the whole communication protocol (e.g. WS-Notification standard) is managed by the Wrapper;
- Keeps an exact copy of the record eliminating the need to reconstruct it when requested. This reduces drastically the impact on the existing source systems.

The two APIs that the Local Wrapper provides are detailed in Table 4.3.

The data received with the *Notify* operation are stored on a local temporal database. To read the received data, the organisations will need either to change the saving procedure redirecting data into their own database or implement interceptors at the database level.

Applications and Services

The Data Sharing Components serve as enabling factor for different potential vertical applications and services. In particular we have developed the EHSR and the BI modules.

API	Description	Input	Output
Notify	Receives metadata from the data processor and save them on a local temporal database table.	$metadata$	$status$
GetRecord	Retrieves the record from the Record Store and returns it to the data processor.	$record_{id}$	$record$

Table 4.3: The list of APIs offered by the Local Wrapper.

Electronic Health and Social Record (EHSR) is represented and implemented by the Event Index that stores the copies of metadata forwarded by the Bus. The Event Index keeps all the event history and provides the basis for an EHR according to Italian guidelines [143]. The guidelines suggest using the ebXML Registry standard that has been adopted by different standards such as IHE [93]. We chose in particular freebXML [78], an open-source reference implementation of ebXML Registry 3.0 standard. Metadata on the Registry are never deleted but instead only marked as deprecated. This design choice provides tractability of actions over patients' medical history.

Business Intelligence (BI) module represents a data consumer and receives *metadata* and filtered (anonymised) *records* that are stored in a Staging Area. After applying extract, transform, and load (ETL) transformations, it saves the data in a Data Warehouse (DWH). An instance of an open source BI framework named SpagoBI [65] produces reports based on the defined KPIs for final users.

The new BI approach that is fed by the developed system and that provides reports over each single service delivered by the plethora of involved organisations is shown in Figure 4.8. This approach reduces the effort for Governance to collect and integrate all the data extracted from single sources as it was done before.

Next section describes how the records are filtered before the delivery to the data consumers.

4.4 Incremental Privacy on Events

Here we describe how data producers define which sensitive information consumers are entitled to access by defined fine-grained filtering policies on the records and based on the purposes of access. For example, in Figure 4.2, the doctor should be able to access all the fields of the record *Rest Home Request* that are necessary for providing assistance (*HealthcareTreatment* purpose), while he/she does not need to access to the fields that indicate the economical status of the patients (*econCoeff* on the record example in Figure 4.4). Figure 4.9 shows an example of such policy.

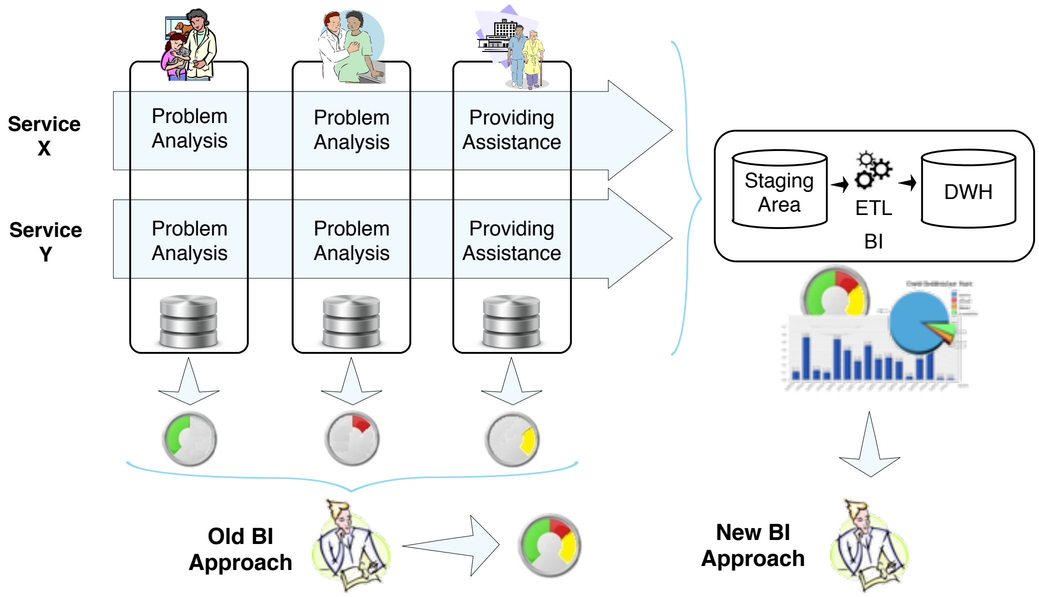


Figure 4.8: The real-time Business Intelligence approach fed by the event-driven architecture.

Figure 4.9 An example of XACML policy defined on the Rest Home Request record for the purpose Healthcare Treatment. The policies are specified using the XACML standard [124] and are defined following the national data protection regulations, which define taxonomies of purposes of use of data [94]. The policy example defines the list of fields (inside the tag *Obligations*) that can be accessed for purpose *HealthcareTreatment* (inside the tag *Actions*). In particular, it disallows the access to fields *econCoeff* and *participCoeff* that indicate the economical situation of the service requester and that are not needed by the doctor for healthcare assistance purpose.

As shown in “The Data-sharing Protocol”, we propose two alternative implementations for the enforcement of the privacy policies: *centralised enforcement* and *decentralised enforcement*. In the centralised enforcement privacy policies are applied by the data processor on the *records* retrieved by the data controller. This approach relieves the data controller from dealing with the enforcement of the privacy policies but it is less privacy-safe. The *records* transferred from the data controller to the data processor contain the sensitive data that is filtered before forwarded to the requestors. Even if the data processor does not persist the data more than the time necessary to apply the policies, there is the potential risk that unfiltered data is intercepted increasing the probability of violations. Furthermore this approach assigns to the data processor the responsibility of granting the correct application of the privacy policies and makes it liable in case of privacy leaks. In some cases, the data processor cannot take charge of such responsibility, and the potential risk of privacy violations could make the approach not completely

```
<Policy ... targetnamespace>
 <Rule RuleId="RestHomeReqPolicy" Effect="Permit">
  <Target>
   <Subjects><Subject>
    <SubjectMatch MatchId="...string-equal">
    <AttributeValue DataType="...string">Doctor</AttributeValue>
    <SubjectAttributeDesignator AttributeId="...role"
     DataType="...string"/>
    </SubjectMatch>
   </Subject></Subjects>
  <Resources><Resource>
   <ResourceMatch MatchId="...string-equal">
    <AttributeValue DataType="... ">RestHomeRequest</AttributeValue>
    <ResourceAttributeDesignator AttributeId="...resource-id"
     DataType="...string"/>
    </ResourceMatch>
   </Resource></Resources>
   <Actions><Action>
    <AttributeValue
      DataType="...string">HealthcareTreatment</AttributeValue>
    <ActionAttributeDesignator AttributeId="...action-id"
     DataType="...string"/>
    </ActionMatch>
   </Action></Actions>
  </Target>
 </Rule>
 <Obligations>
  <Obligation ObligationId="fieldsAvailable" FulfillOn="Permit">
  <AttributeAssignment AttributeId="...field1"  DataType="...string">
  /Record/content/socialStatus</ AttributeAssignment>
  <AttributeAssignment AttributeId="...field2"  DataType="...string">
  /Record/content/probDesc</AttributeAssignment>
 </Obligation></Obligations>
</Policy>
```

Figure 4.9: An example of XACML policy defined on the Rest Home Request record for the purpose Healthcare Treatment.

privacy safe and not approved by the Privacy Guarantor.

In the *decentralised enforcement* the privacy policies are applied by the data controller before the *record* leaves the local repository (i.e. Local Wrapper). This assures that only the data that the requester is authorised to see will be delivered to it. This approach gives more guarantees from the privacy point of view even if it requires a greater effort for the data controller to apply the policies. In any case, the application of the privacy policies could be easily encapsulated in the Local Wrapper module.

In both approaches privacy policies are defined (or chosen from the set of existing ones) by the data producers and are stored on the Visibility Rule Manger that acts as the Policy Retrieval Point. This assures that there exists a single and official place in which privacy policies are maintained simplifying the definition and synchronisation among the parties.

In the current implementation we support only the "hiding" of certain fields but the approach can be easily extended to more advanced privacy policies to mask or to encrypt certain fields (e.g. the SSN). Filtering alone could allow curios consumers to guess the sensitive data from the "missing" values. Consumers could be also able to combine two or more records to mine prohibited information by exploiting data inferences that could arise [55]. Tackling these aspects is a challenging task that requires further investigation in the definition of policies as it opens diverse research questions. Although we are currently working on automating the verification of policy consistency to disallow information leakage [85], in the current system implementation we provide only plain filtering policies. Given that the involved institutions are trusted parties, and that in similar scenarios it is important to deliver the right amount of information [83, 94], our solution has been validated by Privacy Guarantor confirming that it provides sufficient guarantees. In fact, it gives more guarantees than the IHE-XDS profile [93] and the projects that have been based on it [136, 143] and applied worldwide for healthcare domain.

In conclusion, filtering policies in addition to the incremental protocol based on *metadata* and *records* allows us to:

- Conceal sensitive information based on the data producer preferences with a tight control on its distribution;
- Centralise only the *metadata* on the occurrence of events, that is not sensitive in our context and can be stored in the event index with no violation of the privacy laws and directives [94, 143] which disallow data duplication outside the boundaries of its data controller;
- Tune and differentiate the distribution of *metadata* and *records* with an on/off access control for the first and a fine-grained and sub-document access control for the second;
- Manage selective subscriptions and access only to the events of interest minimising the traffic;

Next section report on the validation phases we performed together with the involved organisations in our case study.

4.5 Validation

Here we show how we validated both the process-oriented analysis approach and the system prototype we developed and deployed to be used by the Province of Trento.

4.5.1 The Analysis Approach Validation

As described previously, we modelled the assistance processes and represented them graphically for a better understanding and involvement of stakeholders. In order to obtain a good representation, the model should be very detailed to clarify any ambiguity. To achieve a detailed and satisfactory modelling we performed more than twenty meetings with stakeholders. The overall analysis approach is inspired by collaborative analysis and task oriented analysis techniques [127, 173] that focus on tasks executed by operators. Other methodologies such as ethnographic methodology [32] would require a bigger amount of time and resources to complete the analysis. During each meeting at least two stakeholders were involved and each of the four scenarios were analysed in case they were involved in them. This required investing a considerable amount of time in a deeper analysis making the approach applicable only on a restricted number of scenarios (four in our case). An example of resulting model is show in Figure 4.10.

The formalism we adopted and the whole analysis process enabled us to:

- Discover, model and document only the relevant portions of business processes occurring inside each institution and their inter-relationships and to understand the level of formality of healthcare and socio-assistive processes. Given that each single actor has its own way to operate inspired by best practices of their reference organisation or just by their common sense and past experience, we tried to sketch out the glossary in use, the actions performed, the responsibility (who does what), the exceptions and the constraints (e.g. what should be done to proceed with the next step in another system), input and output data (i.e. events) of an action. The final result should be models usable by non-technical people and by analysts to gather system requirements;

- Capture the data flow and the data format (paper vs. electronic form) to understand how information usually flows and the points of automation to transform paper-based data into "informatised" knowledge that is more "usable";

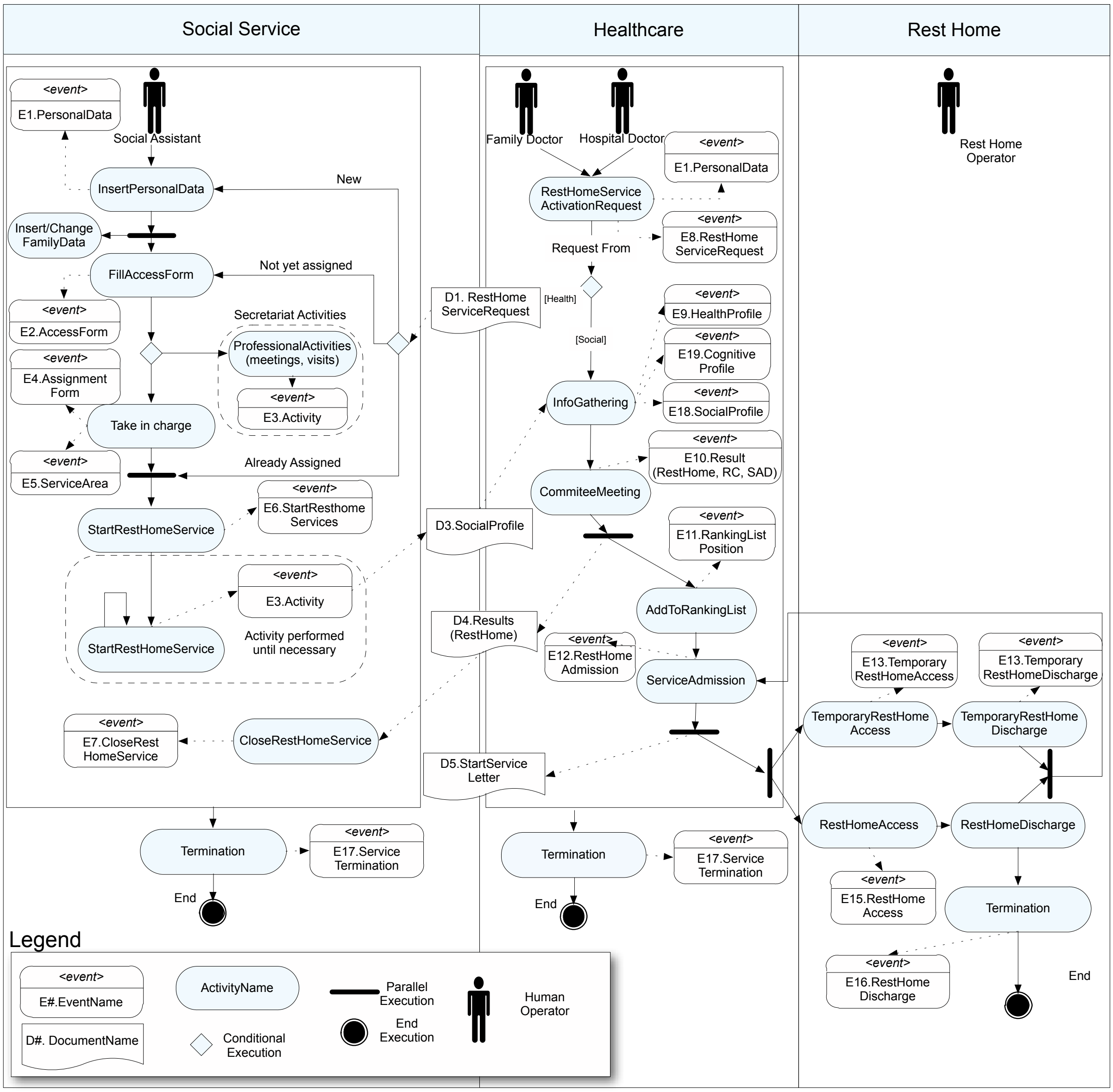


Figure 4.10: Excerpt of artefacts used during the analysis phases to model actors, performed activities and data.

- Isolate events and their contents (fields), domain values, point of generation and frequency of generation, their type, optional or compulsory nature, standardised nomenclature and domain attributes;
- Derive KPI to understand what the user (operators of the healthcare and socio-assistive domain and governing bodies) needs and their expectations from the system.

This approach allowed us to tackle the time limitations and the necessity to cover as many cases as possible and to get a complete picture of the domain. In our analysis we modelled the processes just to identify the events and the conditions for their generation. Indeed, in that way we missed some details but at the same time we simplified the problem to make it tractable.

During the analysis we also tried to standardise the structure of events among different data producers allowing their integration. This implies that the events need to speak the same language, that is, to use a common and shared glossary (e.g. a shared nomenclature of social-health services). We defined about 40 record types. There was no need to deal with the databases at the sources that sometimes have more than hundred of tables, some having more than fifty attributes.

The resulting modelled processes have many exceptions, for example they may start from a social worker but also from the medical staff and each sub-process at the data controllers can end at any point in time (e.g. for rejection of the request or death). To make things even more complex the same information system could be used following a different sequence of interactions.

Overall, even if we identified some limitations of the current approach, it gave us the appropriate tool to involve the stakeholders and complete successfully the analysis phase.

4.5.2 The System Validation

Once finished with the development we defined the deployment plan with the involved stakeholders, which consisted into an on-field two-phase experimentation by:

1. Testing and evaluation of the system in a controlled environment (6 months);

2. Deployment into production to start the evolution from prototype to product at the Province data centre (2 years).

At the time we are writing this thesis, the project is approaching the second year of the second phase. The deployment plan has been defined together with the Province and the involved institutions and refined on the basis of a first round of tests. It will verify if the system is properly dimensioned with regard to the number, size and rate of production of the events (some numbers are shown in Table 4.4), and the IT infrastructures available at the data controllers to verify if they can interact with the platform.

Table 4.4 shows the number of event types that each involved institution is producing and the estimated number of instances generated per year. The estimated required storage is also shown in the last column.

To test the system in the first phase we defined some fictitious citizen profiles to execute a set of complete **request-evaluation-provisioning** assistance processes in the data controllers systems to verify the correct production, routing and consumption of the events. We also computed

Institution	**Number of event types**	**Estimated number of event instances**	**Total event dimension (MB)**
Healthcare Agency	9	14500	56
Welfare Agency	14	74700	251
Local Municipality	7	53100	205
Tele-assistance	7	93985	322

Table 4.4: Estimation of the number of events and data exchanged annually.

a first set of KPIs on the DWH. Although the obtained KPIs from the artificial data were not reflecting the actual real world statistics, they allowed us to test the capability of the infrastructure in feeding the BI module and the usefulness of the indicators identified.

The effort required to join the platform and to share information from the technological point of view was very low since institutions had to implement only a couple of web service invocations. This step nowadays can be performed in few minutes with automated functionalities offered by newer IDEs (Eclipse, NetBeans etc.). We recall that the Local Wrapper that we released to the data controllers, provides the business logic for data storage, requests resolution, waiting for incoming messages and the security protocols. This was one of the key factors for the project success as it minimised considerably the institutions' effort in a scenario in which high learning curve and entry barriers are a deterrent for smaller institutions.

The Visibility Rule Manager and the policies enabled the data producers (assisted by us) to define fine-grained exchange rules over their data. In this way they had a complete control on how and to whom the data will be delivered. This choice respects the main requirement that the data provider is responsible for its own data treatment. The number of policies that a producer will define for each event depends on the requests for subscription to that event. In our scenario the BI module consumes the larger part of events, so the majority of events will have at least one policy to regulate how to feed the DWH. However, there are events that will be consumed by all parties for different purposes and this will imply the need for more policies.

4.6 Lessons Learned and Discussion

Cross-organisation healthcare projects are characterised by unique cultural settings, specific regulations, national-wide and local trends and guidelines. Managing such projects (from analysis to testing phases) requires therefore each time a specific approach that considers the specificity of the contexts and that it is able to involve proactively the stakeholders during the project phases. In Trentino, like in other Italian regions and European countries, the health and social services are provided by different organisations having different interests. One of main critical challenges that needs to be tackled, and that we noticed also in other projects, is about involving organisations that will be asked to spend time and resources for participating initially during the analysis phase, and then, later, in adapting their systems to join the developed infrastructure. The obligations towards the welfare agency for accounting or for providing statistical information could not be sufficient as motivating factors for good engagement to the joint effort and therefore, influencing the project success.

The case study we analyse had as the main objective the reporting, i.e. integrating data from different institutions and generating information about identified KPIs. As such, it seemed to be a classical data-warehousing problem, for which there are by now fairly consolidated techniques. However, from the outset it turned out that our initial assumptions and hopes for a quick solution using data warehousing techniques proved to be wrong, and the problem required considerable research efforts to design a solution that is minimally invasive (from technical and organisational perspective) and that provides strong privacy control. Namely, in such scenarios, centralized integration approaches such as classical data-integration or centralised EHR architectures [89] are not suitable and sometimes forbidden by law. In our case, even the decentralised architectures such as the IHE - XDS [93] that has been successfully applied in many projects [136, 36] have been demonstrated as not suitable “as-is” for the integration of **social** and **health** services. Non-standardised data and assistance processes require a deeper analysis to identify the best trade-off between existing well-established standards and custom solutions to tackle the specific requirements. To address these challenges, we successfully identified different research contributions relatively to the project management aspects (the analysis) and to the applied technology (privacy-aware data-sharing protocol).

The proposed analysis approach that focuses on identifying the events generated during the user-system interaction has been demonstrated to be an exhaustive solution for a complex cross-organisation scenario. This analysis approach identifies which documents can be translated into electronic records, while the internally generated events provide information about the generated data content (list of fields, type, format). However, analysing and automating the cross-organisation document exchanges should be only the first automation step. Starting from it, the cross-organisation processes could be (and should be) further tuned and modified to achieve better performances and better integration among different institutions.

The proposed concise and relatively easily understandable modelling formalism given by the activity diagrams is able to actively involve domain experts to teach to system designers and other domain experts their internal socio-health domain. This has been one of the key factors for achieving a good analysis and therefore for projects success. The domain experts learned also how to perform on their own the analysis of other scenarios and identify the events of interest.

The proposed **privacy-preserving data sharing protocol** minimises the effort for involved institutions to join the platform and to achieve data sharing. The integration approach based on events provides a content independent mechanism to share data, which does not depend on data sources' technology and sources' database schemas.

The sub-document filtering, in conjunction with the incremental protocol based on *metadata* and *records*, provides the sufficiently flexible privacy-preserving techniques to tackle a context that is characterized by several organisations producing and consuming different kinds of data. The filtering policies enable the sharing of records containing socio-assistive, healthcare and financial data without violating the purpose of use principle.

The choice of assigning to the data controllers the task of definition of data filtering policies decentralizes the competences and responsibility. In such way the platform and the organization that manage the platform at runtime, will not have any legal responsibility on the way data is shared among institutions. This simplifies the task of project approval from the Privacy Guarantor.

This approach assumes that data controllers will have the necessary knowledge to define safe filtering policies that do not disclose sensitive information (e.g. forbid to curios consumer to guess the sensitive data from the "missing" values or to combine two records to mine prohibited information). This is a challenging task that requires further investigation in the definition of policies as it opens diverse challenges. These challenges and our work on facilitating the definition of policies and automatic verification of their safety and prevention of information leakage is described in Section 5.

Overall the developed solution provides data-sharing and basic infrastructure for Business Intelligence and Electronic Health and Social Record systems for social and health services. The developed Business Intelligence service gives a global view on the cross-institutional socio-medical processes while the integration approach and the supporting framework represent a new interoperability reference model for institutions involved in health and social assistance. An instance of the developed solution has been successfully applied in a project undertaken by the Province of Trento, Italy.

Chapter 5

Access Control Policy Violation Prevention

Previous chapter describes an access control policy enforcement approach in which each data source defines access control policies locally on its own data (or local schema). In such scenario, however, data sources cannot anticipate data inferences that can arise when data is integrated at the EHR level. Inferences, e.g., using functional dependencies, can allow malicious users to obtain prohibited information by linking multiple queries and thus to violate the local policies.

In this chapter, we describe a framework, *i.e.,* a methodology and a set of algorithms, to prevent such violations. First we identify sets of queries, called violating transactions, that lead to violations if combined based on functional dependencies and then we propose an approach to forbid the execution of those transactions by identifying additional access control rules that should be added to the EHR. We also state the complexity of the algorithms and discuss a set of experiments we conducted by using both real and synthetic datasets. Tests also confirm the complexity and upper bounds in worst-case scenarios of the proposed algorithms.

5.1 Introduction

Data integration offers a convenient way to query different data sources while using a unique entry point (e.g. an EHR system) that is typically called *mediator*. Although this ability to synthesize and combine information maximizes the answers provided to the user, some privacy issues could arise in such a scenario. The authorization policies governing the way data is accessed are defined by each source at local level without taking into consideration data of other sources. In relational and other systems, data constraints or hidden associations between attributes at the mediator level could be used by a malicious user to retrieve prohibited information. One type of such constraints are the *functional dependencies* (FDs). When FDs are combined with authorized information, they may allow the disclosure of some prohibited information. In these cases, there is a need for providing additional mechanisms at the mediator level to forbid the leakage of any prohibited information.

In this work we aim at assisting administrators in identifying such faults and defining additional access control rules at the mediator level to remedy the inference problem. Given a (relational) schema of the mediator, the sources' policies and a set of FDs, we propose a set of algorithms that are able to identify violating transactions. These transactions correspond to sets of queries that violate the sources' policies if used in conjunction with FDs. To avoid the completion of a transaction, and therefore the violation of any source's policy, we propose a query cancellation algorithm that identifies a *minimum set* of queries that need to be forbidden. The identified set of queries is then used to generate additional rules to be added to the existing set of rules of the mediator.

The reminder of the chapter is organized as follows. Section 5.2 provides definitions of the main (technical) concepts we use in the chapter. Section 5.3 introduces a motivating scenario, the integration approach and challenges posed by functional dependencies. In Section 5.4 we describe our methodology. Section 5.5 describes the *detection phase* that identifies the policy violations. Section 5.6 describes the *reconfiguration phase* that deals with flaws identified in the detection phase. Section 5.7 describes the experiments. Finally, we conclude in Section 5.8.

5.2 Preliminaries

Before describing our approach, a number of introductory definitions are needed:

Datalog rule. [1] A (datalog) rule is an expression of the form
$R_1(u_1){:}{-}R_2(u_2), ..., R_n(u_n)$, where $n \geq 1$, $R_1, ..., R_n$ are relation names and $u_1, ..., u_n$ are free tuples of appropriate arities. Each variable occurring in u_1 must also occur in at least one of $u_2, ..., u_n$.

Authorization policy. An authorization policy is a set of authorization rules. An authorization rule is a view that describes the part of data that is *prohibited* to the user. An authorization rule will be expressed using an augmented datalog rule. This augmentation consists in adding a set of predicates characterizing the users to whom the authorization rule applies.

Violating Transaction. A violating transaction T is a set of queries such that if they are executed and their results combined, they will lead to disclosure of sensitive information and thus violating the authorization policy.

Functional Dependency. [107] A functional dependency over a schema R (or simply an FD) is a statement of the form:
$R : X \rightarrow Y$ (or simply $X \rightarrow Y$ whenever R is understood from the context), where $X, Y \subseteq$ schema(R). We refer to X as the left hand side (LHS) and Y as the right hand side (RHS) of the functional dependency $X \rightarrow Y$.

A functional dependency $R : X \rightarrow Y$ is satisfied in a relation r over R, denoted by r $\models R : X \rightarrow Y$, iff $\forall\, t_1, t_2 \in r$ if $t_1[X] = t_2[X]$, then $t_1[Y] = t_2[Y]$.

Pseudo transitivity rule. [107] The pseudo transitivity rule is an inference rule that could be derived from Armstrong rules [14]. This rule states that if $X \rightarrow Y$ and $YW \rightarrow Z$ then $XW \rightarrow Z$.

Without loss of generality we consider functional dependencies having only one attribute in their RHS. A functional dependency of the form $X \rightarrow YZ$ could always be replaced by $X \rightarrow Y$ and $X \rightarrow Z$ by using the decomposition rule [107] which is defined as follows: *if F* $\vdash$ *X* $\rightarrow$ *YZ, then F* $\vdash$ *X* $\rightarrow$ *Y and F* $\vdash$ *X* $\rightarrow$ *Z.*

5.3 Motivating Scenario

We consider a healthcare scenario inspired by our work in Section 3 and 4 while developing the infrastructure for Electronic Health Record (EHR) systems. The EHR in this case represents the mediator which provides mechanisms to share data and to enforce the appropriate authorizations and policies defined by data sources [104]. From that scenario we extract an example that describes how FDs can impact access control and can be challenging to tackle at the mediator level.

We describe the challenges in a **Global as View Integration (GAV)** scenario [106] but the same challenges affect also the Service Oriented integration that we describe in Section 4. Namely, the sensitive information that is released through record requests can be joined, for example, based on the patients' social security number (SSN). In this way a requester can obtain the same data as in the GAV approach by querying all the sources. This is the case for example when data is collected to be used for Business Intelligence. In such case the data is stored in a Data Warehouse that is described usually by a relational schema similarly to the GAV approach.

Here we define an example scenario with three sources. Particularly, we consider the sources S_1, S_2 and S_3 with the following local schemas: $S_1(SSN, \mathit{Diagnosis}, \mathit{Doctor})$ contains the patient social security number (SSN) together with the diagnosis and the doctor in charge of her/him, $S_2(SSN, \mathit{AdmissionT})$ provides the patient admission timestamp, $S_3(SSN, \mathit{Service})$ provides the service to which a patient has been assigned.

The mediator virtual relation, according to the GAV integration approach, is defined by using relations of the sources. We consider a single virtual relation to simplify the scenario but the same reasoning applies for a mediator's schema composed by a set of virtual relations. In our example, the mediator will combine the data of the sources joined over the SSN attribute as shown by rule (5.1).

$$\begin{aligned}&M(SSN, Diagnosis, Doctor, AdmissionT, Service) : -\\ S_1(SSN, Diagnosis, Doctor), &S_2(SSN, AdmissionT), S_3(SSN, Service).\end{aligned} \tag{5.1}$$

Authorization Policies are specified by each source on its local schema and propagated to the mediator. In our example, we assume two categories of users: doctors and nurses. For S_1, doctors can access SSN and $Diagnosis$ while nurses can access either SSN or $Diagnosis$ but not their association (i.e., simultaneously). The rule (5.2) expresses this policy in form of a prohibition.

$$R_1(SSN, Diagnosis) : -S_1(SSN, Diagnosis), role = nurse. \tag{5.2}$$

The other sources allow accessing to their content without restrictions both for doctors and nurses, therefore there are no more authorization rules to specify.

At the Mediator, authorization rules are propagated by the sources aiming at preserving their policies. The propagation can lead to policy inconsistencies and conflicts [57]. These issues are out of the scope of this chapter. In our example there is only one rule defined by S_1 to be propagated at the mediator.

We then assume that at the mediator the following FDs are identified, either manually during the schema definition or by analyzing the data with algorithms such as TANE[91]:

$$(AdmissionT, Service) \to SSN \qquad (F_1)$$

$$(AdmissionT, Doctor) \to Diagnosis \qquad (F_2)$$

F_1 holds because at each service there is only one patient that is admitted at a given time $AdmissionT$. Note that $AdmissionT$ represents the admission timestamp including hours, minutes and seconds. F_2 holds because at a given timestamp, a doctor could make only one diagnosis.

Let see how $\mathcal{FD}$ could be used by a malicious user to violate the rule (5.2). Let us assume the following queries are issued by a nurse: $Q_1(SSN, AdmissionT, Service)$ and then $Q_2(Diagnosis, AdmissionT, Service)$. Combining the results of the two queries and using the functional dependency F_1, the nurse can obtain SSN and $Diagnosis$ simultaneously, which induces the violation of the authorization rule (5.2). To do so, the nurse could proceed as follows: (a) join the result of Q_1 with those of Q_2 on the attributes $AdmissionT$ and $Service$; (b) take advantage of F_1 to obtain the association between SSN and $Diagnosis$.

From now on, we refer to a query set like $\{Q_1, Q_2\}$ as a *violating transaction*. Indeed, both F_1 and F_2 do not hold in any source. They both use attributes provided by different sources. Thus, the semantic constraints expressed by these functional dependencies could not be considered by any source while defining its policy. This example highlights the limitation of the naïve propagation of the policies of the sources to the mediator. In the next section, we propose an intuitive approach for solving this problem.

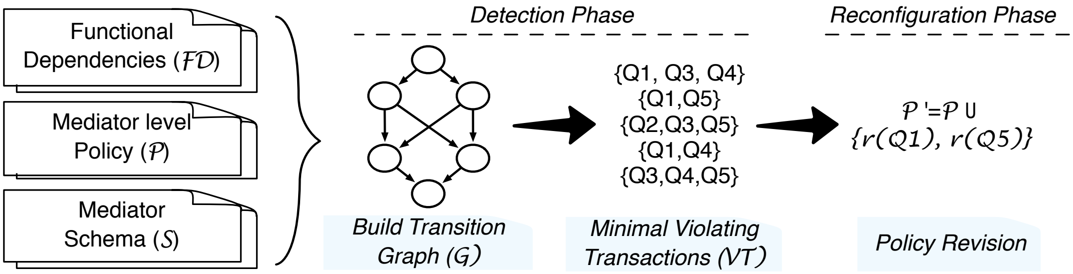

Functional Dependencies ($\mathcal{FD}$)
Mediator level Policy ($\mathcal{P}$)
Mediator Schema ($\mathcal{S}$)
Detection Phase
Reconfiguration Phase
{Q1, Q3, Q4}
{Q1,Q5}
{Q2,Q3,Q5}
{Q1,Q4}
{Q3,Q4,Q5}
$\mathcal{P}$'=$\mathcal{P}$ U {r(Q1), r(Q5)}
Build Transition Graph ($\mathcal{G}$)
Minimal Violating Transactions ($\mathcal{VT}$)
Policy Revision

Figure 5.1: The proposed methodology to identify violating transactions and define additional rules.

5.4 Approach

We propose a methodology that aims at detecting all the possible violations that could occur at the mediator level by first identifying all the violating transactions and then disallowing completion of such violating transactions.

Our approach relies on the following settings: we consider the relational model as the reference model, both user queries and datalog expressions denoting authorization rules (see Section 5.2) are conjunctive queries and the mediator is defined following the GAV (Global As a View) data integration approach. This means that each virtual relation of the mediator is defined using a conjunctive query over some relations of the sources.

Currently we do not consider other types of inferences or background, external or adversarial knowledge that refer to the additional knowledge the user may have while querying a source of information [117, 41]. These aspects are important but they are out of the scope of this thesis.

The proposed methodology, as shown in Figure 5.1, consists of a sequence of phases and steps involving appropriate algorithms. It takes as input a set of functional dependencies ($\mathcal{FD}$), the policy ($\mathcal{P}$) and the schema ($\mathcal{S}$) of the mediator and applies the following phases:

1. ***Detection phase:*** aims at identifying all the violations that could occur using $\mathcal{FD}$. Each of the resulting transactions represents a potential violation. Indeed, as shown previously, the combination of all the queries of a single transaction induces an authorization violation. This phase is performed by the following steps:

    - *Construction of a transition graph ($\mathcal{G}$):* this is done for each authorization rule by using the set of provided functional dependencies ($\mathcal{FD}$).
    - *Identification of the set of Minimal[1] Violating Transactions ($\mathcal{VT}$):* it consists in identifying all the different paths between nodes in $\mathcal{G}$ to generate the set of minimal violating transactions.

[1] The concept of minimality is detailed in Section 5.5.2.

2. ***Reconfiguration phase:*** it proposes an approach to forbid the completion of each transaction in $\mathcal{VT}$ identified in the previous phase. By completion of a transaction we mean issuing and evaluating all the queries of that transaction. A rule is violated only if the entire transaction is completed. This phase modifies/repairs the authorization policy in such a way that no $\mathcal{VT}$ could be completed.

5.5 Detection Phase

In the detection phase we enumerate all the violating transactions that could occur considering the authorization rules as queries that need to be forbidden. The idea is to find all the transactions (i.e., a set of queries) that could match the query corresponding to the authorization rule.

5.5.1 Building the Transition Graph

The aim of the transition graph is to list all the queries that could be derived from an authorization rule using functional dependencies. For each authorization rule we use $\mathcal{FD}$ to derive a transition graph ($\mathcal{G}$) as shown in Figure 5.2. To build $\mathcal{G}$ we resort to Algorithm 1 as follows:

1. Consider the set of attributes of an authorization rule as the initial node.
2. For each FD in $\mathcal{FD}$ that has the RHS attribute inside the current node (starting from the root):
   (a) Create a new node by replacing the RHS attribute of the node with the set of attributes of the LHS of FD.
   (b) Create an edge between the two nodes and label it with F^Q (see Definition 5.5.2) corresponding to the FD that has been used.
3. Apply the same process for the new node.

5.5.2 Identifying Violating Transactions

The set of minimal violating transactions ($\mathcal{VT}$) is constructed as follows. First a path between the initial node (the node representing the authorization rule) and every other node is considered. As shown in Figure 5.2, from this path a transaction (*i.e.,* a set of queries) is constructed. Each query that is used as a label on this path is added to the transaction. Finally, the query of the final node of the path is also added to the transaction. This is done for all nodes and paths in $\mathcal{G}$. Before showing how minimality of the $\mathcal{VT}$ is ensured let us introduce the following definitions.

Algorithm 1: BuildTransitionGraph (BuildG)

```
input  : $r_i$ the rule $r_i \in P$,
         $\mathcal{FD}$ the set of functional dependencies.
output: $\mathcal{G}(V, E)$ the transition graph
$V := \{v(r_i)\}$; // create the root $v$ with the attributes of $r_i$
$W := \{v(r_i)\}$; // add $v$ also to a set $W$ of vertexes to visit
forall the $w \in W$ do
    $W := W - \{w\}$;
    forall the $FD(LHS \rightarrow RHS) \in \mathcal{FD}$ do
        if $RHS \in w$ then                          // $RHS$ is one attribute
            $x := w - \{RHS\} + LHS$; // create new vertex
            if $x \notin V$ then
                $V := V + \{x\}$;
                $W := W + \{x\}$;
            $e := (w, x, LHS + \{RHS\})$; // $e$ is a new edge from $w$ to $x$
                with as transition the attributes $LHS + \{RHS\}$
            if $e \notin E$ then                     // if not already in $E$ add it
                $E := E + \{e\}$;
return $\mathcal{G}(V, E)$ ;
```

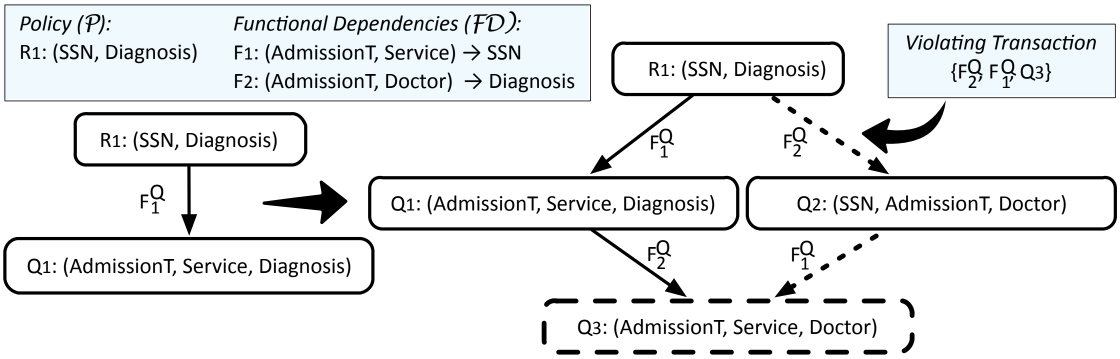

Policy (𝒫):
R1: (SSN, Diagnosis)
Functional Dependencies (ℱ𝒟):
F1: (AdmissionT, Service) → SSN
F2: (AdmissionT, Doctor) → Diagnosis
R1: (SSN, Diagnosis)
F_1^Q
Q1: (AdmissionT, Service, Diagnosis)
R1: (SSN, Diagnosis)
Violating Transaction
{F_2^Q, F_1^Q, Q3}
F_1^Q
F_2^Q
Q1: (AdmissionT, Service, Diagnosis)
Q2: (SSN, AdmissionT, Doctor)
F_2^Q
F_1^Q
Q3: (AdmissionT, Service, Doctor)

Figure 5.2: Graph construction and violating transactions identifications.

Building a query from a functional dependency. Let F be a functional dependency. We define F^Q as the query that projects on all the attributes that appear in F, either in the RHS or in the LHS. For example, let $R(A_1, A_2, A_3, A_4)$ be a relation and let F be the functional dependency $A_1, A_2 \rightarrow A_3$ that holds on R. In this case F^Q is the query that projects on all the attributes that appear in F. F^Q is the query $F^Q(A_1, A_2, A_3){:-}R(A_1, A_2, A_3, A_4)$.

Minimal Query. A query Q is minimal if all its attributes are relevant, that is $\forall Q' \subset Q : Q'$ cannot be used instead of Q in a violating transaction.

Minimal Violating Transaction. A violating transaction T (see Section 5.2) is minimal if: (a) all its queries are minimal, and (b) all its queries are relevant *i.e.* $\forall Q \in T : T \setminus \{Q\}$ is not a violating transaction.

To generate the minimal set of transactions ($\mathcal{VT}$) that is compliant with the definition **??**, we use the recursive Algorithm 2. The initial call to the algorithm is: $\mathcal{VT} := FindVT(\mathcal{G}, root, \emptyset, \emptyset)$

The example in Figure 5.2 contains three nodes Q_1, Q_2 and Q_3 in addition to the initial node R_1. If we apply Algorithm 2, it will generate, for each node Q_i, a transaction containing each F^Q on the path between R_1 and Q_i, and Q_i itself. For example, to generate T_3 that represents the path between R_1 and Q_3, we start by adding each F_i on the path from R_1 to Q_3. Here, F_1 and F_2 are translated into F_1^Q and F_2^Q respectively. Finally, we add Q_3. Thus, we obtain $T_3 = \{F_1^Q, F_2^Q, Q_3\}$. In the example the returned $\mathcal{VT}$ is: $\mathcal{VT} = \{T_1 = \{Q_1, F_1^Q\}, T_2 = \{Q_2, F_2^Q\}, T_3 = \{Q_3, F_1^Q, F_2^Q\}\}$. At this stage we emphasized the fact that $\mathcal{FD}$ could be combined with authorized queries to obtain sensitive information. In our example, this issue is illustrated by the fact that if all the queries of any transaction T_i are issued then the authorization rule $R_1(SSN, Diagnosis)$ is violated. To cope with this problem and prohibit transaction completion, we propose an approach that repairs the set of authorization rules with additional rules in such a way that no violation could occur.

Algorithm 2: FindViolatingTransactions (FindVT)

```
input  : $\mathcal{G}(V, E)$ the transition graph, $v$ current vertex, $c_t$ current path, $\mathcal{VT}$ current set of
         transactions.
output: $\mathcal{VT}$ the set of minimal violating transactions.

foreach $e \in$ outgoing edges of $v$ do
    $t := c_t + e.transition + e.to$ ;
    // $e.transition$ is the set of attributes of the transition
       while $e.to$ is the destination node
    if $\nexists k \in VT \mid k \subseteq t$ then      //if $t$ is minimal with respect to $\forall k \in \mathcal{VT}$
        $\mathcal{VT} := \mathcal{VT} + \{t\}$ ;
        forall the $k \in \mathcal{VT}$ do
            if $t \subseteq k$ then        // if $k$ is not minimal with respect to $t$
                $\mathcal{VT} := \mathcal{VT} - \{k\}$ ;
                // reducing further $VT$
    return $FindVT(\mathcal{G}, e.to, c_t + e.transition, \mathcal{VT})$;
    // recursive call with the $v$ reached by $e$ ($e.to$) by adding
       the $e.transition$ to the current $VT$
```

5.6 Reconfiguration phase

This phase aims at preventing a user from issuing all the queries of a violating transaction. If a user could not complete the execution of all the queries of any violating transaction then no violation could occur.

The reconfiguration phase revises the policy by adding new rules such that no violating transaction could be completed. A naïve approach could be to deny one query for each transaction. Although this naïve solution is safe from an access control point of view, it is not desired from an availability point of view. To achieve a trade off between authorization enforcement and availability, we investigate the problem of finding the minimal set of queries that denies at least one query for each violating transaction. We refer to this problem as *query cancellation problem*. We first *formalize* and characterize the *complexity* of the query cancellation problem for one rule. Then, we discuss the case of a policy (*i.e.,* a set of rules).

5.6.1 Problem formalization

Let $\mathcal{VT} = \{T_1, \ldots, T_n\}$ be a set of minimal violating transactions and let $\mathcal{Q} = \{Q_1, \ldots, Q_m\}$ be a set of queries such that $\forall i \in \{1, \ldots, n\} : T_i \in \mathcal{P}(Q) \smallsetminus \emptyset$. We define the following *Query Cancellation (QC)* recognition (decision) problem as follows:

- **Instance:** a set $\mathcal{VT}$, a set $\mathcal{Q}$ and a positive integer k.
- **Question:** is there a subset $Q \subseteq \mathcal{Q}$ with $|Q| \leq k$ such that $\forall i \in \{1, \ldots, n\} : T_i \setminus Q \neq T_i$? Here, $|Q|$ denotes the cardinality of Q.

Algorithm 3: QueryCancellation

```
input : $\mathcal{VT}$ is the set of minimal violating transactions.
        $Q$ is the set of all the queries that appear in $\mathcal{VT}$
output: $S$ is the set of all solutions
forall the $q \in Q$ do
    if $\forall t \in \mathcal{VT}, t \cap q \neq \emptyset$ then
        $S := S \cup q$ ;
return $S$ ;
```

Thus, the optimization problem, which consists in finding the *minimum number of queries* to be cancelled is called *Minimum Query Cancellation (MQC).*

5.6.2 Problem complexity

In this section, we show the NP-completeness of QC. We propose a reduction from the domination problem in split graphs [31]. In an undirected graph $\mathcal{G} = (V, E)$, where V is the node (vertex) set and E is the edge set, each node *dominates* all nodes joined to it by an edge (neighbors). Let $D \subseteq V$ be a subset of nodes. D is a dominating set of $\mathcal{G}$ if D dominates all nodes of $V \setminus D$. The usual *Dominating Set (DS)*[31] decision problem is stated as follows:

- **Instance:** a graph G and a positive integer k.
- **Question:** does G admits a dominating set of size at most k ?

This problem has been proven to be NP-complete even for split graphs [31]. Recall that a split graph is a graph whose set of nodes is partitioned into a clique C and an independent set I. In other words, all nodes of C are joined by an edge and there is no edge between nodes of I. Edges between nodes of C and nodes of I could be arbitrary.

Theorem 5.6.1. *QC is NP-complete.*

Proof. QC belongs to NP since checking if the deletion of a subset of queries affects all transactions could be performed in polynomial time. Let G be a split graph such that C is the set of nodes forming the clique and I is the set of nodes forming the independent set. We construct an instance QC of query cancellation problem from $\mathcal{G}$ as follows: $\mathcal{Q} = C$, $\mathcal{VT} = I$ and each transaction T_i is the set of queries that are joined to it by an edge in G. We then prove that G admits a dominating set of size at most k if and only if QC admits a subset $Q \subseteq \mathcal{Q}$ of size at most k such that $\forall i \in \{1, \ldots, n\} : T_i \smallsetminus Q \neq T_i$.

Assume QC admits a subset $Q \subseteq \mathcal{Q}$ of size at most k such that $\forall i \in \{1, \ldots, n\} : T_i \smallsetminus Q \neq T_i$. Q is also a dominating set of $\mathcal{G}$. In fact, all nodes of I are dominated since all the transactions are affected by Q and all remaining nodes in the clique C are also dominated since they all are connected with nodes of Q. Assume $\mathcal{G}$ admits a dominating set D of size at most k. Observe that D could be transformed into a dominating set D' of same size and having all its nodes in C. To ensure this transformation it is sufficient to replace all nodes of D that are in I by any of their neighbors in C. Note that the obtained set D' is also a dominating set of $\mathcal{G}$. The subset of queries to be canceled is then computed by setting Q to D'. □

Thus, we can deduce the following:

Corollary 5.6.2. *MQC is NP-hard.*

To generate the set of queries that need to be canceled we use Algorithm 3. It returns all the (candidate) sets of queries that have a non-empty intersection with each violating transaction. We can use different metrics to determine which set to choose. The first metric is the cardinality of the smallest set. Other metrics could be defined by the administrator. Indeed, some queries can be identified as more relevant to the application. In this case, the set of queries to be chosen could be the one that does not contain any relevant query. The minimal set of queries MQ is defined using one of the previous metrics. For each query Q in MQ a new authorization rule is added to prevent from the evaluation of Q.

In our example, the QC algorithm will return three different candidate sets of solutions to be added to $\mathcal{P}$. These sets are: $\{r(Q_1), r(F_2^Q)\}$, $\{r(Q_2), r(F_1^Q)\}$, $\{r(F_1^Q), r(F_2^Q)\}$. If we choose the first candidate set then we will have $\mathcal{P} = \{R_1(SSN, Diagnosis)$, $R_2(AdmT, Service, Diag.), R_3(AdmT, Doctor, Diag.)\}$.

5.6.3 Generalization for a policy

Algorithm 4 deals with query cancellation for the whole policy. We denote by $\mathcal{P}$ the policy (i.e., the set of rules). We denote by N_R the set of new rules that has been generated. The new policy set ($\mathcal{P}$) will be the union of $\mathcal{P}$ and N_R ($\mathcal{P} = \mathcal{P} \cup N_R$). A new rule could generate other new rules and so on until no rule is added. Let N_S be the set of attributes of the mediator schema. Since N_S is finite then the maximum number N_r of rules that could be defined is also finite. Let $N_{\mathcal{P}}$ be the number of rules in $\mathcal{P}$. Let n be the difference between N_r and $N_{\mathcal{P}}$. At each recursive call of the algorithm either no rule has been generated or n decreases since N_r increases. Thus, the algorithm terminates.

Algorithm 4: GenerelizationForPolicy

```
input  : $\mathcal{P}$ the set of authorization rules.
output: $\mathcal{P}$ augmented with new rules.

forall the $r_i \in \mathcal{P}$ do
    $\mathcal{G} := BuildG(r_i)$;
    $\mathcal{VT} := FindVT(\mathcal{G}, root, \emptyset, \emptyset)$;
    $\mathcal{S} := QueryCancellation(\mathcal{VT}, Q)$; // $Q$ is obtained listing $\mathcal{VT}$
    $N_R := \emptyset$; // $N_R$ is the set of new rules
    forall the $q \in \mathcal{S}$ do
        $N_R := N_R \cup \{r(q)\}$; // Generate a new authorization rule $r$ from
            $q$
    if $N_R$ is not empty then
        $N_R := GenerelizationForPolicy(N_R)$;
$\mathcal{P} := \mathcal{P} \cup N_R$;
return $\mathcal{P}$ ;
```

5.7 Validation

We have conducted a number of experiments on real and synthetic datasets to validate each of the steps of our methodology. With synthetic datasets we generated particular configurations (e.g. worst-case scenarios) while with the real datasets (downloaded from the UCI ML Repository [22]) we first extracted $\mathcal{FD}$ by using a well-known algorithm called TANE [91] and then we run our algorithms with sets of rules having different number of attributes (from 2 to 10). We also tested the algorithms on specific subsets of $\mathcal{FD}$ (i.e., 100 and 200 extracted from the Bank dataset) that were not present in real datasets (Sub 1 and 2 in Table 5.7). The source code of the algorithms is released under GPL v3 free software licence and is available at the following address [86].

The reports about measures performed on each dataset shown in Table 5.7 are as follows:

1. **Detection phase:** FD_l is the average number of attributes that appear in $\mathcal{FD}$, $|\mathcal{G}(V)|$ is the number of nodes and $|\mathcal{G}(E)|$ is the number of edges of the generated graph, $BuildG$ is the time in *ms* to build $\mathcal{G}$, $|\mathcal{VT}|$ is the number of generated $\mathcal{VT}$ and $FindVT$ is the time in *ms* to construct $\mathcal{VT}$.

2. **Reconfiguration phase:** $|P'|$ is the number of rules that need to be added to the policy in order to forbid the completion of any transaction in $\mathcal{VT}$.

For each of the tests reported in Table 5.7 we calculated the mean value for 100 different

Dataset desc.		Identified $\mathcal{FD}$		Performed experiments and results					
Name	$\|S\|$	$\|\mathcal{FD}\|$	$\mathcal{FD}_l$	$\|\mathcal{G}(V)\|$	$\|\mathcal{G}(E)\|$	$BuildG$	$\|\mathcal{VT}\|$	$FindVT$	$\|P'\|$
Yeast	8	10	3.88	6	10	5	5	4	7
Chess	20	22	9.14	21	20	3	20	14	21
Breast W.	11	37	4.13	41	165	26	37	65	20
Abalone	8	44	3.79	87	835	60	17	42	23
Sub 1	17	100	4.41	217	1312	193	130	197	54
Sub 2	17	200	4.92	453	8152	1502	1737	16596	263
Bank	17	433	6.47	14788	879241	3826	9137	335607	513

Table 5.1: Features data sets together with results of the experiments.

executions generating rules with a number of attributes ranging from 2 to 10.

While Table 5.7 reports on the approach practicability on real datasets, the graphs in Figures 5.3, 5.4, 5.5 and 5.6 show tests performed on synthetic datasets. Also in this case we run multiple tests while varying parameters that are not subject to the evaluation. In particular, Figure 5.3 shows the relation between the number of nodes and the cardinality of randomly generated $\mathcal{FD}$. We report different tests while varying the number of attributes at the mediator schema. The tests show that by increasing the cardinality of $\mathcal{FD}$, the number of nodes increases very fast until, at a certain point, it starts slowing and approaching its upper bound as expected theoretically. Figure 5.4 shows the relation between the number of nodes and the time needed for building $\mathcal{G}$ with fixed attributes in the mediator schema. As we can see, the time to build $\mathcal{G}$ increases proportionally with respect to the number of nodes. This is mainly because we use binary trees to manage the nodes. The dots in figures represent single executions while the line has been generated using the Spline algorithm [88]. Figure 5.5 reports the performances on identifying $\mathcal{VT}$ from previously built graphs. The time grows proportionally with respect to the number of transactions. With the discovered $\mathcal{VT}$ we extract the additional rules by applying Algorithm 4 to forbid transaction completion. Figure 5.6 shows the relation between the number of transactions and the number of additional rules that are extracted. In particular, at each cycle, we pick as decision metric the new rule that appear more often in $\mathcal{VT}$. We observe that the more FDs are discovered the more rules need to be added. This is due to the fact that more FDs induce more alternatives to policy violations.

The experiments show the practicability of our methodology on different datasets with different characteristics. The approach showed some limitation only when the cardinality of $\mathcal{FD}$ becomes very large (e.g., greater than 1500 for a single relation) being not able to discover transactions in an acceptable amount of time. We believe that this amount of FDs does not represent a typical scenario. Nevertheless, we will further investigate such situations.

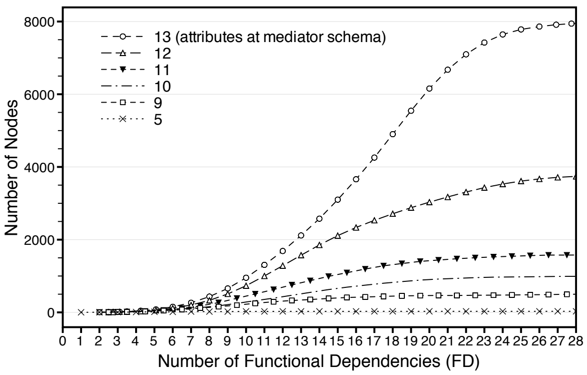


Figure 5.3: Number of nodes with respect to number of FDs.

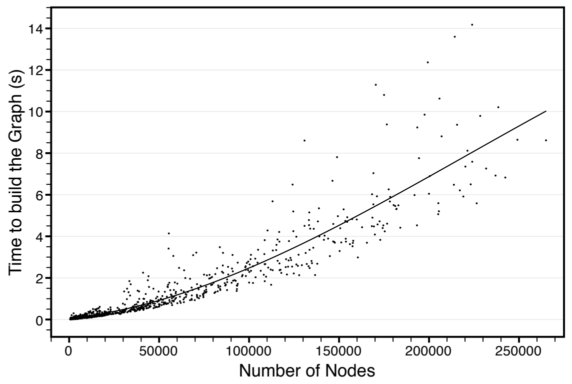


Figure 5.4: Time to build the graph with respect to the number of nodes.

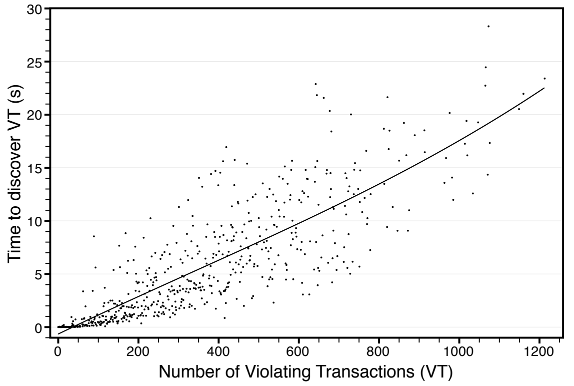


Figure 5.5: Number of VTs and time to identify them.

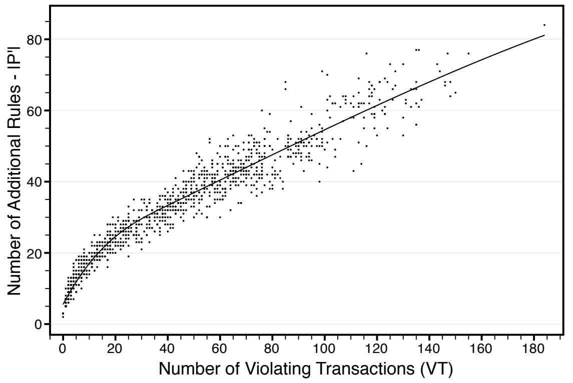


Figure 5.6: Number of VTs and number of added rules.

5.8 Lessons Learned and Discussion

In this work we have investigated the problem of illicit inferences that result from combining semantic constraints with authorized information showing that these inferences could lead to policy violations. To deal with this issue, we proposed an approach to detect the possible violating transactions. Each violating transaction expresses one way to violate an authorization rule. Once the violating transactions are identified, we proposed an approach to repair the policy. This approach aims at adding a minimal set of rules to the policy such that no transaction could be completed. As one alternative to the approach we propose in this chapter, we are analysing the possibility of forbidding queries at runtime in order to avoid $\mathcal{VT}$ completion.

As future work we will extend this approach to partial FDs (i.e., the FDs that do not hold in all tuples but can lead to policy violations). We also plan to investigate other kinds of semantic constraints such as inclusion dependencies and multivalued dependencies. Finally, we could consider other integration approaches such as LAV and GLAV where same issues can arise.

Chapter 6

Conclusion

In this chapter we summarise the thesis contributions and we report the collected feedback and lessons learned from testing the proposed approaches and developed technology. We conclude with final remarks which report the overall lessons learned while working in the healthcare sector.

6.1 Summary of Contributions

We report briefly for each of the three core chapters and for each of the analysed research threads the identified challenges and contributions.

In **Chapter 3** we focus on the analysis of regulatory compliance and organisations' specific requirements aiming at building a cross-organisation and cross-regulation medical record sharing platform. We envision that the developed platform could be offered as a service to healthcare organisations providing the necessary tools to simplify data management (storage and sharing).

Having this vision in mind, we start by showing how the combination of regulatory compliance and organisations' specific requirements represents a complex obstacle to every organisation that deals with privacy sensitive data. We show how regulations and laws vary from country to country by analysing regulatory contexts in Italy and UK.

To help organisations in identifying, defining and executing regulatory compliant data sharing, we proposed a methodology and an execution environment to:

- Capture the sequence of steps that need to be carried out by organisations to define their own security and privacy policies and data sharing processes to conform to high-level regulatory compliance requirements;
- Identify a set of elements, IT components and actors that can be orchestrated by data sharing processes to achieve compliant data sharing;

- Provide an environment for the definition and execution of shared processes and policies to support data, process and policy management.

The overall approach is based on a novel use and customisation of business processes to model the internal business logic of data management operations. Processes serve as a vehicle to achieve high customisation of operations and enable organisations to satisfy their privacy, security and business requirements. CHINO executes in such way data owners' processes and policies when their data is accessed by other organisations. By doing so, while organisations use the same set of interfaces to interact with CHINO, they can achieve regulatory compliance.

Furthermore, the business process based approach allows users to better understand and share their understanding of compliance requirements, and to reason about process definition and process improvements. The visual modelling of business process provides better visibility and transparency on security and privacy rules. This can help at improving the trust and compliance among the participant organisations.

In **Chapter 4** we show how we approached the analysis, design and development of a data sharing protocol and system architecture that preserve privacy while exchanging data among organisations. We show how the proposed algorithms can be used for building Electronic Health Record and a Business Intelligence system for monitoring social and healthcare assistance processes. While Chapter 3 focuses on regulatory compliance, Chapter 4 focuses on developing the underlying technology and privacy policies for sharing sensitive data.

We show that event-based reasoning can greatly simplify the analysis of the organisations' IT systems, facilitating the identification of the data needs among the parties and speeding up the system development. The data sharing protocol and the technology minimises the effort for the institutions to join and use the platform and it grants them full control on the access and distribution of their data.

We give to the institutions the possibility to define fine-grained policies and control the dissemination of data based on the purposes of use. The data sharing protocol limits the disclosure of data by releasing less sensitive information (metadata) and only when needed, delivering more sensitive data (records) that has been properly filtered.

An instance of the developed solution has been successfully applied in a project undertaken by the Province of Trento, Italy. The integration approach and the supporting framework represent a new interoperability reference model for institutions involved in health and social assistance.

The contributions shown in Chapter 4 have been applied in Chapter 3 and in building the CHINO platform providing the main components, the data sharing protocol and the concept of using events to share data.

Chapter 5 analyses challenges related to the prevention of access control policies violations in data integration scenarios. We show how defining access control policies in integration scenarios such as the one described in Chapter 4, can be challenging due to the presence of illicit inferences that result from combining semantic constraints (e.g. given by functional dependencies) with authorised information. We analyse in particular how functional dependencies can lead to policy violations. To deal with this issue, we proposed an approach to detect the violations and to repair the set of policies defined by data sources by adding new policies.

These contributions, when incorporated in the technology proposed in Chapter 4 for the definition of sub-document access control policies, can lead to safer privacy policies for EHR systems. Next section describes how the contributions of this dissertation have been validated and the limitations that have been identified.

6.2 Validations, Limitations and Future Work

We report the main validation phases by following their chronological order to better present to the reader the maturity level of contributions, their adoption in practice and their limitations.

Our work on the threads covered by this thesis starts with what is described in Chapter 4 and the development of a data sharing and integration platform for social and health assistance. The research effort is part of an innovation and development project undertaken by the province of Trento. The reported research contributions have been validated within the project with real customers and stakeholders. The developed technology, after being revised by a development unit of a participant company, is now gradually being adopted by institutions delivering care in the Province of Trento. Overall, its design choices have been validated successfully demonstrating their suitability for the considered scenario.

One aspect that needs to be further improved is about the access control policies defined at sub-document level to control the disclosure of data. Namely, although the technology has been successfully validated, as shown in Chapter 5, the policy definition strategy can reveal weakness and bring to policy violations in case of presence of data dependencies. This issue needs to be further analysed at runtime by applying the technology shown in Chapter 5. As shown in Chapter 5, this is a challenging task since semantic constraints can be of different types (e.g. functional, inclusion or multivalued dependencies). In our work we consider only functional dependencies and therefore, the work needs to be extended by analysing if other types of inferences can generate the same issues. Furthermore, inferences can also hold partially (i.e. not holding in all tuples or records) and thus requiring the definition of acceptable thresholds under which the inferences are not considered risky. Other challenges related to this aspect can consist of identifying semantic constraints without full access rights to the whole database of all data sources. In addition, database content changes over time so there is the need to identify semantic constraints by starting from the initial phase in which there is no data and then at runtime when the data is generated and changes.

Therefore, tacking these challenges is extremely complex, which leads to the adoption of approximate solutions to satisfy the typical constraints for industrial projects (e.g. time and resources).

After validating the work described in Chapter 4 within the Italian regulatory context, we analysed the HIPAA regulations and data sharing best practices in US, and we discovered that the proposed solution was not suitable and it was requiring significant modifications to data sharing protocol and policies. Starting from this observation, we analysed challenges related to cross-regulation data sharing and built the CHINO platform. Overall the work collects experiences and lessons learned while developing data management solutions in Italy and US (by collaborating with a team from HP Labs in Palo Alto, US).

To test the CHINO methodology and technology validity, we performed different tests:

- The CHINO platform has been validated by integrating it with a popular medical record system called OpenMRS and by defining data sharing processes and policies according to Italian, UK and HIPAA regulations and best practices.

- The validity of internal components and the data sharing protocol suitability in healthcare have been validated also within the contributions of the Chapter 4. Namely, they have been developed and tested within the project described in Chapter 4 before starting the CHINO development.

- The usability of the CHINO modelling framework and the ability of business analysts and developers to model business processes to conform to regulatory policies, have been validated with a group of nine business process experts.

- By involving privacy experts we verified that the proposed tools and features in CHINO provide the guarantees and ability to satisfy current regulatory policies and best practices under Italian legislation.

We are currently adopting the CHINO methodology and concepts of the CHINO technology, to model and automate assistance processes in a socio-health scenario within an innovation and development project that involves many public and private institutions and caregivers providing assistance to elders [49]. The adoption of the methodology in IT projects as an analysis approach gave us some encouraging feedback and confirmations about its validity. The visual modelling of data management aspects has been demonstrated as an intuitive tool to involve stakeholders.

Although the CHINO technology has been validated successfully, the natural extension of the work would be a cloud-based platform for medical record sharing offered to a global market. It represents a challenging and promising deployment scenario, yet rare to identify in practice and in projects on which it can be tested and validated with real users.

6.3 Final Remarks

This thesis reports our research results and lessons learned while developing medical record sharing in multi-organisation settings. The presented work is inspired by research questions identified while working on industrial and research projects and the overall approach is driven by pragmatism and the aim of applying the research outcomes in ongoing and future activities and projects.

Some of the challenges we identified in such context are related to: regulations complexity, many actors having different roles and accessing data for different purposes, existence of variety of technical standards, technology that still requires research efforts, assistance process complexity, mix of private and public institutions, organisations business interests being not completely in line with data sharing, organisations focused on assistance instead of developing IT infrastructure, and organisations having limited resources. For example, the presence of a multitude of actors such as hospitals, social-assistance public and private providers, accounting offices, personal doctors, and many more, makes the healthcare an everyday life scenario characterised by extremely complex challenges related to privacy and access control over data (e.g. as shown in Chapter 5).

The most important lesson learned while working in such context is that every IT project needs to be approached differently. Namely, managing and developing integration and data sharing solutions requires state-of-the-art project management methodology and technology, but at the same time pragmatism. In such a scenario, sometimes the most efficient and state-of-the-art solutions simply could not work due to the impact on current information systems used by organisations or due to the required time and costs to deploy it. There is usually the need to preserve the institutions business interests.

Therefore, given the specificity and the complexity of the healthcare sector a conclusion is that **context matters**. As shown in Chapter 5, sometimes there can be issues which require too much effort to be tackled and sometimes a perfectly safe technology from security and privacy points of views do not (still) exists. Therefore, depending on the context, even not completely privacy safe solutions can be appropriate and applicable and moreover, as it has been the case with the scenario described in Chapter 4, the proposed technology can introduce significant improvements to the national-wide architectures and protocols [93, 143]. Namely, compared to our solutions, the proposed national-wide standards do not apply any fine-grained access control at sub-document level to manage the disclosure of medical records. Instead, we introduce stronger privacy guarantees at integration layer to deliver to data consumers only the data they need to perform their tasks.

The market evolution toward software as a service and cloud-based models inspired our work in Chapter 3 on building the CHINO platform. This delivery model could have high impact in situations where small agencies use information systems to provide a limited set of services. In such situation lightweight and less intrusive solutions need to be proposed to enable them to sustain the required effort and meet the requirements imposed by integration solution. This is also important if we consider the high complexity of information systems used by bigger institutions from which data need to be extracted. In such cases there is the need to propose a lightweight integration approach that avoids analysing their internal system and database structures. Although the potential deployment of CHINO in cloud-based environments requires further analysis of other research issues such as security and scalability, it represents an interesting possibility following the market trends and it represents one of the main directions in which our future research activity should move towards.

Bibliography

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of databases*, volume 8. Addison-Wesley Reading, 1995.

[2] Activiti. Activiti bpm platform. `http://activiti.org/`. Accessed: 2013-12-20.

[3] Douglas Adams. The Hitchhiker's Guide to the Galaxy, 1978.

[4] Gagan Aggarwal, Mayank Bawa, Prasanna Ganesan, Hector Garcia-Molina, Krishnaram Kenthapadi, Rajeev Motwani, Utkarsh Srivastava, Dilys Thomas, and Ying Xu. Two can keep a secret: A distributed architecture for secure database services. *CIDR 2005*, 2005.

[5] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Hippocratic databases. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 143–154. VLDB Endowment, 2002.

[6] Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju. *Web Services: Concepts, Architecture and Applications*. Springer Verlag, 2004.

[7] Anne H. Anderson. A comparison of two privacy policy languages: `EPAL` and `XACML`. In *SWS '06: Proceedings of the 3rd ACM workshop on Secure web services*, pages 53–60, New York, NY, USA, 2006. ACM.

[8] Apache. Apache servicemix. `http://servicemix.apache.org/`. Accessed: 2013-12-20.

[9] Ajit Appari and M Eric Johnson. Information security and privacy in healthcare: current state of research. *International journal of Internet and enterprise management*, 6(4):279–314, 2010.

[10] Giampaolo Armellin, Dario Betti, Stefano Bussolon, Annamaria Chiasera, Manuela Corradi, and Jovan Stevovic. From PHR to NHR? An UCD challenge. In *International workshop on Personal Health Record*, Trento, Italy, 2011.

[11] Giampaolo Armellin, Dario Betti, Fabio Casati, Annamaria Chiasera, Gloria Martínez, and Jovan Stevovic. Privacy preserving event driven integration for interoperating social and health systems. In *Proceedings of the 7th VLDB Conference on Secure Data Management*, SDM'10, pages 54–69, Berlin, Heidelberg, 2010. Springer-Verlag.

[12] Giampaolo Armellin, Dario Betti, Fabio Casati, Annamaria Chiasera, Gloria Martínez, Jovan Stevovic, and Tefo James Toai. Event-driven privacy aware infrastructure for social and health systems interoperability: CSS Platform. In *ICSOC*, pages 708–710, 2010.

[13] Giampaolo Armellin, Annamaria Chiasera, Ivan Jureta, Alberto Siena, and Angelo Susi. Establishing information system compliance: An argumentation-based framework. In *Research Challenges in Information Science (RCIS), 2011 Fifth International Conference on*, pages 1–9. IEEE, 2011.

[14] William Ward Armstrong. Dependency structures of data base relationships. In *IFIP Congress'74*, pages 580–583, 1974.

[15] Article 29 Data Protection Working Party. Working Document on the processing of personal data relating to health in Electronic Health Records (EHR), wp 131, 2 2007.

[16] Article 29 Data Protection Working Party. The future of privacy: Joint contribution to the consultation of the european commission on the legal framework for the fundamental right to protection of personal data, wp 168, 12 2009.

[17] Article 29 Data Protection Working Party. Opinion 15/2011 on the definition of consent, wp 187, 7 2011.

[18] Article 29 Data Protection Working Party. Working Document 01/2012 on epSOS, wp 189, 1 2012.

[19] Article 29 Data Protection Working Party. Opinion 3/2013 on purpose limitation, wp 203, 4 2013.

[20] ASTM International. CCR - Continuity of Care Record. `http://enterprise.astm.org/filtrexx40.cgi?+REDLINE_PAGES/E2369.htm/`. Accessed: 2013-12-20.

[21] Ahmed Awad, Rajeev GorÃl', James Thomson, and Matthias Weidlich. An iterative approach for business process template synthesis from compliance rules. In Haralambos Mouratidis and Colette Rolland, editors, *Advanced Information Systems Engineering*, volume 6741 of *Lecture Notes in Computer Science*, pages 406–421. Springer Berlin Heidelberg, 2011.

[22] Kevin Bache and Moshe Lichman. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2013. Accessed: 2013-12-20.

[23] Paolo Balboni and Milda Macenaite. Privacy by design and anonymisation techniques in action: Case study of ma3tch technology. *Computer Law and Security Review*, 29(4):330 – 340, 2013.

[24] Jakob E Bardram and Thomas R Hansen. Peri-operative coordination and communication systems: A case of cscw in medical informatics. In *Proc. Of the CSCW 2010 workshop on CSCW Research in Healthcare: Past, Present and Future*, 2010.

[25] Adam Barth, Anupam Datta, John C. Mitchell, and Helen Nissenbaum. Privacy and contextual integrity: Framework and applications. In *Proc. of the 2006 IEEE Symposium on Security and Privacy*, pages 184–198, Washington, DC, USA, 2006. IEEE Computer Society.

[26] Adam Barth, John Mitchell, Anupam Datta, and Sharada Sundaram. Privacy and utility in business processes. *Computer Security Foundations Symposium, IEEE*, 0:279–294, 2007.

[27] T Beale, S Heard, D Kalra, and D Lloyd. openehr - architecture overview. *OpenEHR Foundation*, pages 1–79, 2007.

[28] Kevin Beaver and Rebecca Herold. *The Practical Guide to Hipaa Privacy and Security Compliance*. Auerbach Publications, 2004.

[29] Rachel K. E. Bellamy, Thomas Erickson, Brian Fuller, Wendy A. Kellogg, Rhonda Rosenbaum, John C. Thomas, and Tracee Vetting Wolf. Seeing is believing: designing visualizations for managing risk and compliance. *IBM Syst. J.*, 46(2):205–218, April 2007.

[30] Elisa Bertino, Sushil Jajodia, and Pierangela Samarati. Supporting multiple access control policies in database systems. In *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, pages 94–107. IEEE, 1996.

[31] Alan A. Bertossi. Dominating sets for split and bipartite graphs. *Inf. Process. Lett.*, 19(1):37–40, September 1984.

[32] Hugh Beyer and Karen Holtzblatt. *Contextual Design: Defining Customer-Centered Systems*, volume 1. Morgan Kaufmann, 1998.

[33] Travis D Breaux, Matthew W Vail, and Annie I Anton. Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. In *Requirements Engineering, 14th IEEE International Conference*, pages 49–58, 2006.

[34] Kathryn Breininger and Mary McRae. ebxml registry tc v3.0. Technical report, OASIS, 2005.

[35] Alexander Brodsky, Csilla Farkas, and Sushil Jajodia. Secure databases: Constraints, inference channels, and monitoring disclosures. *Knowledge and Data Engineering, IEEE Transactions on*, 12(6):900–919, 2000.

[36] Canada Health Infoway. `http://www.infoway-inforoute.ca/`. Accessed: 2013-12-20.

[37] Marco Casassa Mont. Dealing with privacy obligations: Important aspects and technical approaches. In Sokratis Katsikas, Javier Lopez, and GÃijnther Pernul, editors, *Trust and Privacy in Digital Business*, volume 3184 of *Lecture Notes in Computer Science*, pages 120–131. Springer Berlin / Heidelberg, 2004.

[38] Daniele Catteddu. Cloud computing: Benefits, risks and recommendations for information security. In Carlos Serro, Vicente Aguilera Daz, and Fabio Cerullo, editors, *Web Application Security*, volume 72, pages 17–17. Springer Berlin Heidelberg, 2010.

[39] Ann Cavoukian. Privacy in the clouds. *Identity in the Information Society*, 1(1):89–108, 2008.

[40] Ann Cavoukian. Privacy by design. *Take the Challenge. Information and Privacy Commissioner of Ontario, Canada*, 2009.

[41] Bee-Chung Chen, Kristen LeFevre, and Raghu Ramakrishnan. Privacy skyline: privacy with multidimensional adversarial knowledge. In *Proceedings of the 33rd international conference on Very large data bases*, VLDB '07, pages 770–781. VLDB Endowment, 2007.

[42] Carolina Ming Chiao, Vera Künzle, and Manfred Reichert. Towards object-aware process support in healthcare information systems. In *4th International Conference on eHealth, Telemedicine, and Social Medicine (eTELEMED 2012)*, pages 227–236. IARIA, February 2012.

[43] Annamaria Chiasera, Fabio Casati, Daniel Florian, and Yannis Velegrakis. Engineering privacy requirements in business intelligence applications. In *Proc. of the 5th VLDB workshop on Secure Data Management*, pages 219–228, Auckland, New Zealand, 2008. Springer Verlag.

[44] Michele Chinosi and Alberto Trombetta. Integrating Privacy Policies into Business Processes. In *WOSIS*, pages 13–25. INSTICC Press, 2008.

[45] Shih-Chien Chou and Chun-Hao Huang. An extended xacml model to ensure secure information access for web services. *J. Syst. Softw.*, 83(1):77–84, 2010.

[46] Richard Chow, Philippe Golle, Markus Jakobsson, Elaine Shi, Jessica Staddon, Ryusuke Masuoka, and Jesus Molina. Controlling data in the cloud: outsourcing computation without outsourcing control. In *Proc. of the 09 ACM workshop on Cloud computing security*, CCSW '09, pages 85–90, New York, NY, USA, 2009. ACM.

[47] Valentina Ciriani, Sabrina De Capitani Di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Keep a few: Outsourcing data while maintaining confidentiality. In *ESORICS 2009*, pages 440–455. Springer, 2009.

[48] Joinup ISA European Commission. The cartella socio sanitaria (social welfare and healthcare data folder) - CSS project. `https://joinup.ec.europa.eu/software/css/description`, 2011. Accessed: 2013-12-20.

[49] SUITCASE Project Consortium. Suitcase project. `http://www.smartcrowds.net/#!suitcase/c1yo9`. Accessed: 2013-12-20.

[50] Manuela Corradi, Annamaria Chiasera, Giampaolo Armellin, and Jovan Stevovic. Understanding how people work: experiences in improving healthcare practices in italy. In *Workshop on Coordination, Collaboration and Ad-hoc Processes (COCOA'10)*, Palo Alto, CA, USA, 2010.

[51] Douglas Crockford. Introducing JSON. `json.org`. Accessed: 2013-12-20.

[52] J. Damasceno, F. Lins, R. Medeiros, B. Silva, A. Souza, D. Aragao, P. Maciel, N. Rosa, B. Stephenson, and J. Li. Modeling and executing business processes with annotated security requirements in the cloud. *Web Services, IEEE Intern. Conf*, 0:137–144, 2011.

[53] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. A fine-grained access control system for xml documents. *ACM Trans. Inf. Syst. Secur.*, 5(2):169–202, 2002.

[54] Nicodemos Damianou, Naranker Dulay, Emil Lupu, and Morris Sloman. The ponder policy specification language. In *Policies for Distributed Systems and Networks*, pages 18–38. Springer, 2001.

[55] Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Assessing query privileges via safe and efficient permission composition. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 311–322. ACM, 2008.

[56] Harry S. Delugach and Thomas H. Hinke. Wizard: A database inference analysis and detection system. *Knowledge and Data Engineering, IEEE Transactions on*, 8(1):56–66, 1996.

[57] Sabrina De Capitani di Vimercati and Pierangela Samarati. Authorization specification and enforcement in federated database systems. *J. Comput. Secur.*, 5(2):155–188, March 1997.

[58] DigitPA. Sistema pubblico di connetivita'. `http://www.digitpa.gov.it/cad/sistema-pubblico-connettivit-spc`. Accessed: 2013-12-20.

[59] Asuman Dogac, Gokce B. Laleci, Yildiray Kabak, Seda Unal, Sam Heard, Thomas Beale, Peter L. Elkin, Farrukh Najmi, Carl Mattocks, David Webber, and Martin Kernberg. Exploiting ebxml registry semantic constructs for handling archetype metadata in healthcare informatics. *Int. J. Metadata Semant. Ontologies*, 1(1):21–36, 2006.

[60] Robert H Dolin, Liora Alschuler, Sandy Boyer, Calvin Beebe, Fred M Behlen, Paul V Biron, and Amnon Shabo Shvo. The hl7 clinical document architecture, release 2. *Journal of the American Medical Informatics Association*, 13(1):30–39, 2006.

[61] Yitao Duan, John Canny, and Justin Zhan. P4p: Practical large-scale privacy-preserving distributed computation robust against malicious users. In *Proceedings of the 19th USENIX Conference on Security*, USENIX Security'10, pages 14–14, Berkeley, CA, USA, 2010. USENIX Association.

[62] Marlon Dumas and Arthur HM Ter Hofstede. UML activity diagrams as a workflow specification language. In *UML 2001 ÑThe Unified Modeling Language. Modeling Languages, Concepts, and Tools*, pages 76–90. Springer, 2001.

[63] eHealth Initiative. Results from survey on health data exchange 2013 - the challenge to connect. `http://www.ehidc.org/2013-hie-survey-results`. Accessed: 2013-12-20.

[64] Marco Eichelberg, Thomas Aden, Jörg Riesmeier, Asuman Dogac, and Gokce B. Laleci. A survey and analysis of electronic healthcare record standards. *ACM*, 37:277–315, December 2005.

[65] Engineering. SpagoBI. `http://www.spagobi.org/`. Accessed: 2013-12-20.

[66] Enterprise Java XACML Implementation. `http://code.google.com/p/enterprise-java-xacml/`. Accessed: 2013-12-20.

[67] EU - ICT PSP Work Programme. epSOS - European eHealth Project. `http://www.epsos.eu/`. Accessed: 2013-12-20.

[68] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, 2003.

[69] European Parliament and Council. Directive 95/46/EC: Directive on protection of individuals with regard to the processing of personal data and on the free movement of such data. `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML`, 1995. Accessed: 2013-12-20.

[70] European Parliament and Council. Directive 2011/24/eu: Directive on the application of patients' rights in cross-border healthcare, 2011.

[71] European Parliament and Council. Article 33 of the general data protection regulation, data protection impact assessment (DPIA), 2012.

[72] European Parliament and Council. Proposal for a regulation on the protection of individuals with regard to the processing of personal data and on the free movement of such data (general data protection regulation, 2012.

[73] Wenfei Fan, Shuai Ma, Yanli Hu, Jie Liu, and Yinghui Wu. Propagating functional dependencies with conditions. *Proceedings of the VLDB Endowment*, 1(1):391–407, 2008.

[74] Csilla Farkas and Sushil Jajodia. The inference problem: a survey. *ACM SIGKDD Explorations Newsletter*, 4(2):6–11, 2002.

[75] Niels Ferguson and Bruce Schneier. *Practical cryptography*, volume 141. Wiley New York, 2003.

[76] Office for Civil Rights. HIPAA, medical privacy - national standards to protect the privacy of personal health information, 2000.

[77] Free Software Foundation. Gnu general public license. `http://www.gnu.org/licenses/gpl.html`. Accessed: 2013-12-20.

[78] freebXML. OASIS ebXML registry reference implementation project (ebxmlrr). `http://ebxmlrr.sourceforge.net/`. Accessed: 2013-12-20.

[79] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv.*, 42(4):14:1–14:53, June 2010.

[80] Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. In *Proc. of the 30th Annual int. conf. on Theory and applications of cryptographic techniques: advances in cryptology*, EUROCRYPT'11, pages 129–148, Berlin, Heidelberg, 2011. Springer-Verlag.

[81] Alessio Giori. Master degree thesis: Design, development and validation of a methodology and platform for compliance-aware medical record management., 2013.

[82] David G. Gordon and Travis D. Breaux. Managing multi-jurisdictional requirements in the cloud: towards a computational legal landscape. In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, CCSW '11, pages 83–94, New York, NY, USA, 2011. ACM.

[83] Italian Privacy Guarantor. Personal data protection code. Legislative Decree no. 196 dated 30 June 2003, 2003.

[84] Italian Privacy Guarantor. Italian privacy guarantor neswletter february 14 2013. http://www.garanteprivacy.it/web/guest/home/docweb/-/docweb-display/docweb/2256479, 2013.

[85] Mehdi Haddad, Mohand-Said Hacid, and Robert Laurini. Data integration in presence of authorization policies. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 92–99. IEEE, 2012.

[86] Mehdi Haddad, Jovan Stevovic, Annamaria Chiasera, Yannis Velegrakis, and Mohand-Said Hacid. Access control for data integration project homepage, http://disi.unitn.it/%7estevovic/acfordi.html, 2013. Accessed: 2013-12-20.

[87] Mehdi Haddad, Jovan Stevovic, Annamaria Chiasera, Yannis Velegrakis, and Mohand-Saïd Hacid. Access control for data integration in presence of data dependencies. In S.S. Bhowmick et al., editor, *DASFAA*, pages 203–217. Springer-Verlag, 2014.

[88] Trevor Hastie and Robert Tibshirani. *Generalized additive models*. London: Chapman & Hall, 1990.

[89] Richard Hillestad, James Bigelow, Anthony Bower, Federico Girosi, Robin Meili, Richard Scoville, and Roger Taylor. Can electronic medical record systems transform health care? Potential health benefits, savings, and costs. *Health Affairs*, 24(5):1103, 2005.

[90] Jörg Hoffmann, Ingo Weber, and Guido Governatori. On compliance checking for clausal constraints in annotated process models. *Information Systems Frontiers*, 14(2):155–177, April 2012.

[91] Ykä Huhtala, Juha Kärkkäinen, Pasi Porkka, and Hannu Toivonen. Tane: An efficient algorithm for discovering functional and approximate dependencies. *The Computer Journal*, 42(2):100–111, 1999.

[92] IEEE Standards Association. The 802.3-2008 standard, 2008.

[93] IHE. IHE - Integrating the Healthcare Enterprise XDS Profile. `http://www.ihe.net/profiles/`. Accessed: 2013-12-20.

[94] Italian Data Protection Authority. Guidelines on the electronic health record and the health file, 2009.

[95] Jin Jing, Ahn Gail-Joon, Covington Michael J, and Zhang Xinwen. Toward an access control model for sharing composite electronic health records. *Information Systems J.*, 2008.

[96] Marwane El Kharbili, Ana Karla A. de Medeiros, Sebastian Stein, and Wil M. P. van der Aalst. Business process compliance checking: Current state and future challenges. In *MobIS'08*, pages 107–113, 2008.

[97] Jyrki Kivinen and Heikki Mannila. Approximate inference of functional dependencies from relations. *Theor. Comput. Sci.*, 149(1):129–149, September 1995.

[98] Nadzeya Kiyavitskaya, Nicola Zeni, Travis D. Breaux, Annie I. Antón, James R. Cordy, Luisa Mich, and John Mylopoulos. Extracting rights and obligations from regulations: toward a tool-supported process. In *Proc. of the 22nd IEEE/ACM intern. conf. on Automated software engineering*, ASE '07, pages 429–432, New York, NY, USA, 2007. ACM.

[99] Anthony Klug. Calculating constraints on relational expression. *ACM Transactions on Database Systems (TODS)*, 5(3):260–290, 1980.

[100] Anthony Klug and Rod Price. Determining view dependencies using tableaux. *ACM Transactions on Database Systems (TODS)*, 7(3):361–380, 1982.

[101] Alzbeta Krausova, Fabio Massacci, and Ayda Saidane. Legal patterns implement trust in it requirements: When legal means are the “best" implementation of it technical goals. In *Proc. of the 2009 2nd Intern. Workshop on Requirements Engineering and Law*, RELAW '09, pages 33–38, Washington, DC, USA, 2009. IEEE Computer Society.

[102] Antonio Kung, J Freytag, and Frank Kargl. Privacy-by-design in its applications. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, pages 1–6. IEEE, 2011.

[103] Peifung E. Lam, John C. Mitchell, and Sharada Sundaram. A formalization of hipaa for a medical messaging system. In *Proc. of the 6th Intern. Conf. on Trust, Privacy and Security in Digital Business*, TrustBus '09, pages 73–85, Berlin, Heidelberg, 2009. Springer-Verlag.

[104] Meixing Le, Krishna Kant, and Sushil Jajodia. Cooperative data access in multi-cloud environments. In *Data and Applications Security and Privacy XXV*, volume 6818, pages 14–28. Springer Berlin Heidelberg, 2011.

[105] Richard Lenz and Manfred Reichert. It support for healthcare processes - premises, challenges, perspectives. *Data Knowl. Eng.*, 61(1):39–58, April 2007.

[106] Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246. ACM, 2002.

[107] Mark Levene and George Loizou. *A guided tour of relational databases and beyond.* Springer Verlag, 1999.

[108] Jun Li, Sharad Singhal, Ram Swaminathan, and Alan H. Karp. Managing data retention policies at scale. *IEEE Transactions on Network and Service Management*, 9(4):393–406, 2012.

[109] Jun Li, Bryan Stephenson, Hamid R. Motahari-Nezhad, and Sharad Singhal. Geodac: A data assurance policy specification and enforcement framework for outsourced services. *IEEE Transactions on Services Computing*, 4:340–354, 2011.

[110] Ruopeng Lu, Shazia Sadiq, and Guido Governatori. Compliance aware business process design. In Arthur Hofstede, Boualem Benatallah, and Hye-Young Paik, editors, *Business Process Management Workshops*, volume 4928 of *Lecture Notes in Computer Science*, pages 120–131. Springer Berlin Heidelberg, 2008.

[111] Bo Luo, Dongwon Lee, Wang-Chien Lee, and Peng Liu. Qfilter: fine-grained run-time xml access control via nfa-based query rewriting. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 543–552, New York, NY, USA, 2004. ACM.

[112] Emil C Lupu and Morris Sloman. Conflicts in policy-based distributed systems management. *Software Engineering, IEEE Transactions on*, 25(6):852–869, 1999.

[113] Kenneth Mandl, William Simons, William Crawford, and Jonathan Abbett. Indivo: a personally controlled health record for health information exchange and communication. *BMC Medical Informatics and Decision Making*, 7(1):1–25, 2007.

[114] H. Mannila and K.J. Räihä. Algorithms for inferring functional dependencies from relations. *Data & Knowledge Engineering*, 12(1):83–99, 1994.

[115] RS Mans, Nick C Russell, Wil MP van der Aalst, Piet JM Bakker, Arnold J Moleman, and Monique WM Jaspers. Proclets in healthcare. *J. of Biomedical Informatics*, 43(4):632–649, August 2010.

[116] Yague Mariemma. Survey on xml-based policy languages for open environments. *Journal of Information Assurance and Security*, pages 11–20, 2006.

[117] David J. Martin, Daniel Kifer, Ashwin Machanavajjhala, Johannes Gehrke, and Joseph Y. Halpern. Worst-case background knowledge in privacy. In *ICDE*, pages 126–135, 2007.

[118] Michael Menzel, Ivonne Thomas, and Christoph Meinel. Security requirements specification in service-oriented business process management. In *Availability, Reliability and Security, 2009. ARES'09. International Conference on*, pages 41–48. IEEE, 2009.

[119] Brenda M. Michelson. Event-driven architecture overview event-driven soa is just part of the eda. *Patricia Seybold Group*, 2006.

[120] Microsoft. HealthVault. `www.microsoft.com/healthvault`. Accessed: 2013-12-20.

[121] Amalia R Miller and C Tucker. Privacy protection and technology diffusion: The case of electronic medical records. *Management Science*, 55(7):1077–1093, 2009.

[122] Zoran Milosevic, Shazia Sadiq, and Maria Orlowska. Translating business contract into compliant business processes. In *Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference*, EDOC '06, pages 211–220, Washington, DC, USA, 2006. IEEE Computer Society.

[123] CNN Money. Obama's big idea: Digital health records. `http://money.cnn.com/2009/01/12/technology/stimulus_health_care/`, 2009. Accessed: 2013-12-20.

[124] Tim Moses. extensible access control markup language tc v2.0 (xacml). Technical report, OASIS, 2005.

[125] Alain Mouttham, Liam Peyton, Ben Eze, and Abdulmotaleb Saddik. Event-driven data integration for personal health monitoring. *J. of Emerging Technologies in Web Intelligence*, 1(2), 2009.

[126] Mulesoft. Mule ESB. `mulesoft.com`. Accessed: 2013-12-20.

[127] Michael J Muller. Participatory design: the third space in hci. *Human-computer interaction: Development process*, pages 165–185, 2003.

[128] Municipality of Trento. Regulations for the protection of personal data of the municipality of trento. `http://www.comune.trento.it/`, 2007. Accessed: 2013-12-20.

[129] Municipality of Trento. Operational guidelines to privacy. `http://www.comune.trento.it/`, 2009. Accessed: 2013-12-20.

[130] Sarah Nait-Bahloul, Emmanuel Coquery, and Mohand-Said Hacid. Authorization policies for materialized views. In *Information Security and Privacy Research*, pages 525–530. Springer, 2012.

[131] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *CoRR*, abs/cs/0610105, 2006.

[132] NHS - National Program for IT (NPfIT). Output based specification for integrated care records service, 2002.

[133] NHS-UK. NHS Connecting for Health. `http://www.connectingforhealth.nhs.uk/`. Accessed: 2013-12-20.

[134] Qun Ni, Elisa Bertino, Jorge Lobo, Carolyn Brodie, Clare-Marie Karat, John Karat, and Alberto Trombeta. Privacy-aware role-based access control. *ACM Transactions on Information and System Security (TISSEC)*, 13(3):24, 2010.

[135] Qun Ni, Alberto Trombetta, Elisa Bertino, and Jorge Lobo. Privacy-aware role based access control. In *SACMAT '07: Proceedings of the 12th ACM symposium on Access control models and technologies*, pages 41–50, New York, NY, USA, 2007. ACM.

[136] NICTIZ-AORTA. AORTA the dutch national infrastructure. `http://www.ringholm.de/docs/00980-en.htm`. Accessed: 2013-12-20.

[137] Anil Nigam and Nathan S Caswell. Business artifacts: An approach to operational specification. *IBM Systems Journal*, 42(3):428–445, 2003.

[138] OASIS. Ws-notification v1.3. `https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn`, 2006. Accessed: 2013-12-20.

[139] Object Management Group. Business Process Model and Notation (BPMN) version 2.0. http://www.omg.org/spec/BPMN/2.0/, April 2011. Accessed: 2013-12-20.

[140] C. Obry, J. H. Jahnke, A. Onabajo, and W. Schafer. Enabling privacy in cross-organisational information mediation " an application in health care. In *Proc. of the 11th Intern. Workshop on SW Technology and Eng. Practice*, pages 155–163, Washington, DC, USA, 2003. IEEE Computer Society.

[141] U.S. Department of Health and Human Services. Consultations and transfers of care detailed use case, 2008.

[142] Department of Health (UK). Confidentiality: Nhs code of practice, 2003.

[143] Italian Ministry of Innovation and Technology. Infse: Technical infrastructure for electronical health record systems, 2010.

[144] Province of Trento. Social and housing services policies, social services in trentino. http://www.trentinosociale.it/. Accessed: 2013-12-20.

[145] Congress Of The United States Congressional Budget Office. Evidence on the costs and benefits of health information technology, 2008.

[146] OpenMRS. http://openmrs.org/. Accessed: 2013-12-20.

[147] Avner Ottensooser, Alan Fekete, Hajo A Reijers, Jan Mendling, and Con Menictas. Making sense of business process descriptions: An experimental comparison of graphical and textual notations. *Journal of Systems and Software*, 85(3):596–606, 2012.

[148] Silvia Panzarasa, Silvana Quaglini, Giuseppe Micieli, Simona Marcheselli, Mauro Pessina, Corrado Pernice, Anna Cavallini, and Mario Stefanelli. Improving compliance to guidelines through workflow technology: implementation and results in a stroke unit. *Studies in Health Technology and Informatics*, 129(2):834, 2007.

[149] Gibbons Patricia, Arzt Noam, et al. Coming to terms: Scoping interoperability for health care. *Health Level Seven, EHR Interoperability Work Group*, 2007.

[150] Tao Peng, Marco Ronchetti, Jovan Stevovic, Annamaria Chiasera, and Giampaolo Armellin. Business process assignment and execution from cloud to mobile. In *International Workshop on Emerging Topics in Business Process Management*, pages 54–69, 2013.

[151] Practice Fusion - Free Web-based Electronical Health Record. www.practicefusion.com. Accessed: 2013-12-20.

[152] Jan C Recker and Alexander Dreiling. Does it matter which process modelling language we teach or use? an experimental study on understanding process modelling languages without formal education. In *18th Australasian Conference on Information Systems*. University of Southern Queensland, 2007.

[153] Nick Russell, Wil MP van der Aalst, Arthur HM Ter Hofstede, and Petia Wohed. On the suitability of UML 2.0 activity diagrams for business process modelling. In *Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling-Volume 53*, pages 95–104. Australian Computer Society, Inc., 2006.

[154] Shazia Sadiq, Guido Governatori, and Kioumars Namiri. Modeling control objectives for business process compliance. In *Proc. of the 5th Intern. Conf. on Business process management*, BPM'07, pages 149–164, Berlin, Heidelberg, 2007. Springer-Verlag.

[155] Ravi S Sandhu, Edward J Coyne, Hal L Feinstein, and Charles E Youman. Role-based access control models. *Computer*, 29(2):38–47, 1996.

[156] Peter Schaar. Privacy by design. *Identity in the Information Society*, 3(2):267–274, 2010.

[157] Helen Sharp. *Interaction design*. Wiley.com, 2003.

[158] Amit P. Sheth and James A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.*, 22(3):183–236, September 1990.

[159] Alberto Siena, Giampaolo Armellin, Gianluca Mameli, John Mylopoulos, Anna Perini, and Angelo Susi. Establishing regulatory compliance for information system requirements: An experience report from the health care domain. In *Conceptual Modeling–ER 2010*, pages 90–103. Springer, 2010.

[160] Roberts Simon and Prendergast David. Community supports for ageing. *Technology Research for Independent Living (TRIL) Centre*, 5(2):169–202, 2009.

[161] Linas Simonaitis, Anne Belsito, Gary Cravens, Changyu Shen, and J Marc Overhage. Continuity of care document (ccd) enables delivery of medication histories to the primary care clinician. In *AMIA Annual Symposium Proceedings*, volume 2010, page 747. American Medical Informatics Association, 2010.

[162] Spring Framework. `http://www.springsource.org/`. Accessed: 2013-12-20.

[163] Jovan Stevovic, Eleonora Bassi, Alessio Giori, Fabio Casati, and Giampaolo Armellin. Enabling privacy by design in medical records sharing. In *In Pre-proceedings of Reforming Data Protection: The Global Perspective*. Springer Netherlands, 2014.

[164] Jovan Stevovic, Jun Li, Hamid Motahari-Nezhad, Fabio Casati, and Giampaolo Armellin. Business process management enabled compliance-aware medical record sharing. *Int. J. of Business Process Integration and Management*, 6(3):201 – 223, 2013.

[165] Jovan Stevovic, Jun Li, Hamid Motahari-Nezhad, Fabio Casati, Giampaolo Armellin, and Bilal Farraj. Compliance aware cross-organization medical record sharing. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 772–775, 2013.

[166] Jovan Stevovic, Jeff Shrager, Alban Maxhuni, Gregorio Convertino, Iman Khaghanifar, and Randy Gobbel. Adding individual patient case data to the melanoma targeted therapy advisor. In *PervasiveHealth*, pages 85–88, 2013.

[167] Tzong-An Su and Gultekin Özsoyoglu. Data dependencies and inference control in multilevel relational database systems. In *IEEE S. on Sec. and Privacy*, pages 202–211. IEEE Computer Society, 1987.

[168] Edgar Tello-Leal, Omar Chiotti, and Pablo David Villarreal. Process-oriented integration and coordination of healthcare services across organizational boundaries. *Journal of medical systems*, 36(6):3713–3724, 2012.

[169] The Direct Project. `http://directproject.org/`. Accessed: 2013-12-20.

[170] Beale Thomas. openehr architecture and model, 2008.

[171] MB Thuraisingham. Security checking in relational database management systems augmented with inference engines. *Computers & Security*, 6(6):479–492, 1987.

[172] New York Times. A face is exposed for aol searcher no. 4417749. `http://select.nytimes.com/gst/abstract.html?res=F10612FC345B0C7A8CDDA10894DE404482`, 2006. Accessed: 2013-12-20.

[173] Leslie Gayle Tudor, Michael J Muller, Tom Dayton, and Robert W Root. A participatory design technique for high-level task analysis, critique, and redesign: The card method. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 37, pages 295–299. SAGE Publications, 1993.

[174] United States Department of Health and Human Services. Health information technology for economic and clinical health (HITECH) act. `http://www.healthit.gov/`, 2009. Accessed: 2013-12-20.

[175] Wil MP van Der Aalst, Arthur HM Ter Hofstede, Bartek Kiepuszewski, and Alistair P Barros. Workflow patterns. *Distributed Parallel Databases*, 14(1):5–51, July 2003.

[176] W3C. Xmlschema. `http://www.w3.org/2001/XMLSchema`, 2001. Accessed: 2013-12-20.

[177] W3C. Simple Object Access Protocol (SOAP). `http://www.w3.org/TR/soap12-part1/`, 2007. Accessed: 2013-12-20.

[178] D.Z. Wang, X.L. Dong, A.D. Sarma, M.J. Franklin, and A. Halevy. Functional dependency generation and applications in pay-as-you-go data integration systems. In *12th International Workshop on the Web and Databases*, 2009.

[179] Hui (Wendy) Wang and Ruilin Liu. Privacy-preserving publishing data with full functional dependencies. In *Proceedings of the 15th international conference on Database Systems for Advanced Applications - Volume Part II*, DASFAA'10, pages 176–183. Springer-Verlag, 2010.

[180] David Webber and Anthony Dutton. Understanding ebxml, uddi, xml/edi. Technical report, XML Global Technologies Inc., 2000.

[181] Jens H Weber-Jahnke and Christina Obry. Protecting privacy during peer-to-peer exchange of medical documents. *Information systems frontiers*, 14(1):87–104, 2012.

[182] Rigo Wenning, Matthias Schunter, Lorrie Cranor, Massimo Marchiori, et al. The platform for privacy preferences 1.1 (P3P1. 1) specification. *W3C Working Group Note*, 2006.

[183] Eric Yuan and Jin Tong. Attributed based access control (ABAC) for web services. In *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on*. IEEE, 2005.

[184] Rui Zhang and Ling Liu. Security models and requirements for healthcare application clouds. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 268–275. IEEE, 2010.