PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI - University of Trento

Privacy elicitation and utilization in distributed data exchange systems

Annamaria Chiasera

Co-Advisor:

Prof. Fabio Casati

Università degli Studi di Trento

Co-Advisor:

Prof. Yannis Velegrakis

Università degli Studi di Trento

April 2012

Abstract

Recently we are assisting to the advent of many data integration projects to allow the cooperation of systems in the more disparate fields (healthcare, finance, education, public security). This trend responds to the increasing needs of data to monitor, compare, correlate and analyse the distributed business processes managed by different institutions and companies for different purposes. As the availability of data in electronic form increases, the risk of improper use of sensitive information is raising also.

In this thesis work we focus on the problem of realising an infrastructure for the data and application integration of systems in the healthcare domain. Our solution is compliant with the privacy regulations, reconciling the visibility requirements of the institutional data consumers with the needs of control and protection of the data subjects. It is an event-based solution which allows to capture the processes going on between the systems to be integrated in a way that is flexible, decoupled and adherent to reality. Our solution enables the sharing of very fine-grained pieces of information to a wide range of consumers still allowing the producers to control who can see what and for what purposes. The architecture minimizes the transit of sensitive information and controls the distribution of events and of their content at a very fine-grained level. In this thesis work we take into account also the impact of the proposed solution on the existing systems ensuring to minimize the effort of companies and institutions in adopting the infrastructure.

As legal privacy regulations are most of the time quite distant from unambiguous IT requirements we investigate the problem of privacy constraints elicitation. Typically privacy constraints are defined manually with a tedious procedure by the IT experts based on the desiderata of the users. This approach is not always yielding the best results as designers lacks the domain knowledge required to produce complete, meaningful and not over-constraining privacy requirements. We believe the user holds the knowledge of the domain and of the data that is necessary to define privacy constraints at the right level of granularity. In particular, we provide a novel approach to privacy constraints elicitation based on the interaction with the user. Our approach derives from high level indications given by the user a concise definition of the privacy constraints directly applicable to the underlying database. Such constraints can be used to further restrict the data values that can appear in a report.

Keywords

privacy, EHR, data sampling

To my mother Emiliana

Acknowledgements

I want to thank my advisors Yannis and Fabio for their guidance and support during this long journey. Without them I would have never completed this work. I am also gratefully to Giampaolo for his patience and his guidance. He teaches me a lot about work and life and I hope in a long collaboration. I want to thank also Fausto and Marco for giving me the opportunity to pursue my PhD with a joint company-university program.

Thanks to Dario, Cesare, Cristiana, Cristina, Gloria, Ioana, Jovan, Juan, Leandro, Manuela, Massimo, Michele, Tao and Tefo for being good colleagues, and most importantly, good friends. I want to thank the GPI company, the University of Trento, FBK and the welfare and healthcare unit of the Province of Trento and in particular Alberto, Angelo, Anna, Cinzia, Daniela, Gabriella, Giovanna, Florian and Loredana for giving me the opportunity to work with them on research as also on real projects.

Thanks to Ida, Anelia, Angela, Salvatore and Sucheta for their wise suggestions and support in all these years.

A special thank to my family, my father Tarcisio, Alessandro, Albina, Jan Philipp, Antonietta, Mariano and Sandra for being with me also in the most difficult moments. I will never say thank enough to Mario for his support and his faith in me, my partner in life and strongest supporter of this work. Without him I will never find the motivation and the commitment to arrive at that point.

This thesis is dedicated to a great woman which is looking at this work from the sky: my mother Emiliana. Please, keep always looking from there.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Modern healthcare relies on computerized means to share data among different organizations and institutions with the intent to create a complete medical history of the patients. This poses also concerns on the sharing of sensitive data and the rights patients can exercise on their data that falls under the umbrella of privacy law [88]. This thesis considers different aspects related to privacy in healthcare systems and proposes an architecture for the integration of healthcare systems and tools to automate the generation and compliance checking of privacy policies.

In Section 1.1 we present the problem and why current information integration approaches are not enough. We present a case study in which the developments of this thesis was applied and its challenging aspects. In Section 1.2 we show the research areas and our contributions. Finally Section 1.3 outlines the structure of the whole thesis.

1.1 Motivation

Recently we are assisting to a strong commitment of the public administrations toward e-government projects focused mainly on the dematerialization of the administrative processes, to monitor, control and trace the clinical and assistive processes [18]. Initially, the computerization in healthcare was limited to a single data controller (e.g. a hospital or a nursing home) developing their own management systems for medical data called Health Files (HFs) [88]. Even if there are many healthcare professionals involved (e.g. doctors, nurses, social workers) they belonged to one single healthcare body acting as data controller.

An HF improves the quality of care as it simplifies the management and improves the accessibility of healthcare data but is limited only to one institutions. Typically, healthcare processes span multiple institutions (hospital, nursing home, no profit organizations) and to support their integration or, as a minimum, to monitor their execution, some degree

of integration to cooperate and exchange information among multiple IT systems is necessary.
Electronic Health Records (EHR) have been introduced to overcome this limitation (Figure 1.1). An EHR combines the medical data distributed in many health files at different healthcare bodies typically in the same region o geographical area. An EHR is accessed by healthcare professionals of different institutions and the data has different data controllers. As defined in [88] EHR's data is "*generated by the various health care professionals and/or bodies providing medical care to that individual over time*". An EHR greatly improves the quality of care [13] as the medical staff can easily access all the past clinical events of the user (e.g. diagnosis, hospitalization records, emergency care).
When sensitive information (about health and economical state of a person, habits, political and religious opinions) are distributed, accessed, stored and used to develop reports like in an EHR, *privacy* becomes one of the most important properties that need to be preserved.

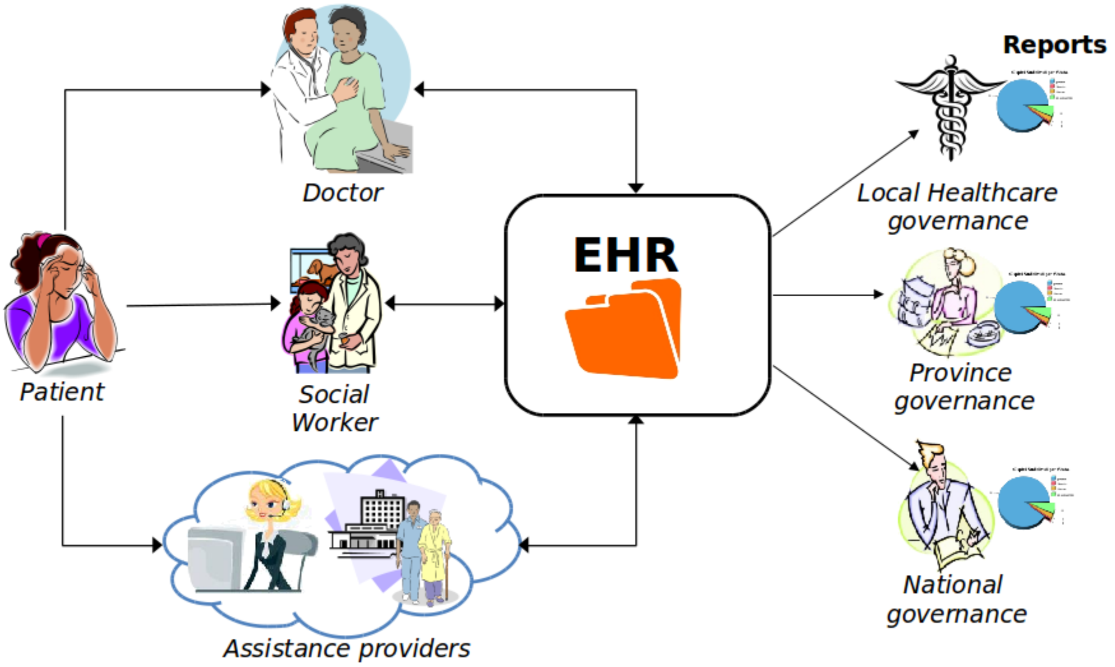


Figure 1.1: Health and Social services interoperability: the EHR.

In this thesis we identified the risks of privacy violations and the properties the system should exhibit to avoid them. We show how to develop a data integration solution that meets these privacy requirements, throughout the different development phases. In doing that, we depart from classical data warehousing solutions which are not applicable in this context as they impose some design decisions that are not allowed by the privacy regulations, they lack flexibility and they are too monolithic for a distributed, dynamic and continuously changing environment like with an EHR. We propose an EHR system based on Service Oriented Architecture (SOA) and on Event Driven Architecture (EDA) that proves to reduce the effort to understand and model the data dependencies among institutions and their data needs abstracting from their internal data structures. We exploit the potentiality of the pub/sub approach to guarantee full visibility on the data to all the EHR's partners still preserving the privacy of the individuals with privacy constraints enforced at the event broker level.

Notice how for an HF, privacy is - in most cases - an issue of limited importance as data stays within a single organization for healthcare purpose. But for EHR, multiple data controllers and many users co-exists with different roles and purpose of access. In addition to "common sense" rules in managing sensitive data, companies have to obey to laws, more or less precisely stating the obligations and restrictions in handling sensitive data. Conservative solutions with a coarse grained access control to the patients data are too restrictive as they tend to hide too much to the medical staff, even data that in the end are not so sensitive. The problem is that, in order to define the right procedures to manage and protect the data, a certain knowledge on the privacy regulations, on what the data represents and on which data should be considered sensitive or not is required. Developers does not have such knowledge that instead is expertise of the user.
In our work we want to exploit the knowledge of the users in understanding which data in a database should be protected thought they cannot work directly on the database, nor use formal languages to express constraints on it. Furthermore, the database is usually a complex one, comprising millions of records, and it is not feasible for the user to go through all of it and tell us what is wrong.
In this dissertation we propose an approach to discover how privacy is perceived by the users. We derive the privacy constraints from indications given by the user of the form "*I don't want to see this value in any report*". Such approach tries to minimize the effort made by the user by selecting samples of the original database the user is able to handle in a reasonable time and that don't exceed a single screen dimension. We apply data mining techniques to identify the more representative elements of the database to be shown to the user with the aim to capture quickly the constraints the user desires to define. The output of the privacy elicitation phase is a declarative representation of the privacy constraints that can be used to further restrict the data values that are allowed to appear in a report. In doing that we search for the more concise representation of the privacy constraints to guarantee they are general enough to remain valid even upon changes in the original data and such that they are as "actionable" (i.e. verifiable, testable) as possible. Our privacy constraints can be expressed as select-project queries so that it is easy to

find privacy violations. Our system guarantees that only the necessary information is accessed by authorized users by filtering out the data that is sensitive according to some privacy constraints defined directly by the data controllers. In order to be approved by the institutions in charge of verifying the correct application of the privacy regulations all the design decisions and the protocols employed in an EHR solution should be properly documented and motivated. In this thesis work we applied also some modelling formalisms to trace which requirements have been implemented in the system to answer to specific privacy laws.

1.1.1 Context

The solutions presented in this thesis have been applied and validated in a large integration project called CSS (Cartella Socio-Sanitaria[1], Social and Health Record) undertaken by the Autonomous Province of Trento (Italy) with a dozen of partners from the IT sector, the public administration and the healthcare services. The project aimed at monitoring healthcare and social processes across the different government and healthcare institutions in the region of Trentino, Italy. Trentino was used as a pilot region in Italy, and the results will next be applied at the national level.

In this scenario users are required to provide documents containing almost the same information to different offices, even if they belong to the same administration (e.g. the financial and welfare departments) [18]. This is obviously quite frustrated for citizens as well as for the caregivers since they need to spend precious working hours to re-type (almost the same) data taken from documents generated by other entities into their IT systems, which also increases the risk to do mistakes.

The complexity of the processes involved and the lack of coordination badly affect the quality of the services delivered as perceived by the stakeholders:

- *patients* experience delays and lack of visibility on the progress of their requests;
- *organizations* waste time to enter information from documents to their IT systems;
- *the governance* lacks a comprehensive interoperability infrastructure to monitor services delivered and resources consumed analyzed by temporal, demographic, and territorial dimensions, to identify needs and trends in social, economical and medical phenomena.

The last point is particularly relevant for the governance as it does not have visibility on the whole business processes going on along the different systems. It has only a partial vision on the single systems sending periodically accounting information and statistical

[1] http://www.trentinosociale.it/index.php/Il-nuovo-welfare/E-welfare/Progetto-Cartella-Socio-Sanitaria-CSS

data but at a very low rate (typically each 3 months). Furthermore, it takes time also to analyse the data in order to combine the partial results obtained from a single source into a global perspective covering the province as a whole. What happens is that the final reports are available one month (or even more) after the data is sent from the sources and they are typically incomplete or they contain mistakes as they are collected manually. The consequence is that the governance will plan the investments in healthcare and welfare services with unreliable data and reports that are at least 4 months old.

In addition to that, an agreed privacy management procedure among the institutions involved is missing, and when data is moved from a system to another, they end up in a sort of "*no mans' land*" in which it is not clear which privacy constraints to enforce. This prevents the sharing of data, unless in sporadic circumstances, which requires further effort to compensate for the lack of a systematic way to prove the adherence to the privacy regulations.

In this scenario it is easy to have unintentional privacy breaches, as the data owners (doctors, social workers and third party assistance providers) that collect the data from the patients do not have any fine-grained control on the data they exchange or send to the governing body [18]. Very often, either they make the data inaccessible (over-constraining approach) or they release to the others more data than what is needed in conflict with the principle of minimal usage [57].

Furthermore, as there is no central controller of the data access requests there is no way to trace how data is used by whom and for what purpose and to be able to answer to auditing inquiry by the privacy guarantor or by the data subjects.

In this thesis we show how to devised an interoperability solution to collect, share and distribute in a more timely way the data on healthcare and social services with a tight control on the sharing of sensitive data. Specifically the goal is to allow the **cooperation** of applications from different agencies (both public and private) to provide high quality clinical and socio-assistive services to the citizens in compliance with the privacy protection code.

1.2 Research Issues and Contributions

Application Integration poses many well-known problems in terms of bridging the different protocols and data models of the various systems involved.

Especially for cross-domain integration projects in healthcare at regional and national level the institutions to be integrated are very high in number and very heterogeneous in nature. Furthermore, institutions progressively join the integrated process monitoring ecosystem, so that an additional challenge lies in how to facilitate the addition of new

institutions.
From a data privacy perspective, these kinds of scenarios present tough challenges, such as:

- how to make it simple for all the various data sources to define the privacy constraints and enforce them. Simplicity here is the key as otherwise the complexity of the information system, and in some cases also of the law, make the problem already hard to tackle, and unless the privacy model remains really simple it is very hard to understand which information is protected and to what extent.
- how to achieve patient/citizen empowerment by supporting consent collection at data source level (opt-in, opt-out options to share the events and their content)
- how to allow monitoring and tracing of the access requests (who did the request and why/for which purpose?)
- how to support the testability and auditability of privacy requirements: owners of data sources often require that the privacy rules they are asked to define can be tested and audited so that they can be relieved of the responsibility of privacy breaches.

All the above cases involve 1) the adoption of data and application integration techniques, 2) the definition of contracts, as precise as needed, that define the work to be done and the expected results, and 3) information flowing across several organizations, with significant privacy issues involved (which also need to be subject to contractual agreements).
In this thesis we study the application integration problems from a privacy perspective and particularly we focus on two problems. One is to find an architecture for the lightweight integration of IT systems in the social and medical domains that is compliant to privacy regulations. The other problem is in enabling the users to control their data by defining in a simple way which data is private and which is not. In the rest of the section we show the complexity of these problems and why current solutions are not applicable.

1.2.1 Privacy-aware EHR Design

The realization of an EHR system is quite challenging because, in addition to the classical issues present in all the information integration systems, the privacy regulations impose additional requirements on the way data are processed and, consequently, on the design and behaviour of the integration system. Particularly, an EHR system should: prevent unauthorized access to sensitive information with access control mechanisms, reduce the replication of sensitive data in many places to trace easily any access attempt, minimize the quantity of sensitive data transferred among different systems, give to the data owners full control in deciding which data can be seen by whom and for which purposes.

A comprehensive EHR solution satisfying the requirements mentioned above is not available in literature.

The challenges are not only technical but are also related to the domain complexity and heterogeneity of the different institutions and actors involved (see Figure 1.1). It is difficult to keep all of them focused in a short-term joint effort as they have different expectations, goals and backgrounds from sociology, medicine, financial management, IT. 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 organization, let alone integrate dozens of them, developed and managed by different institutions.

If we go for a more flexible integration patterns based on SOA and EDA we have to guarantee that the data dissemination capability of the system due to the use of these technologies is controlled and customized according to the needs of the data producers to avoid uncontrolled privacy leaks.

Another challenge is to identify which data can be shared electronically among different systems, and then, how to make these systems interoperable requiring minimal effort and limited changes to existing solutions, a feature which could benefit especially small companies and organizations (since they usually lack the economical and technical resources to do the integration on their own).

1.2.2 Privacy Policy Definition

Even if the system is designed in compliance with the privacy regulations, it may be possible that sensitive information is exposed (even by fault) in a report derived from the integrated data or an event released by a source system. The problem is to understand which data can be released without violating any privacy regulations, and to understand it hopefully before the reports are published to avoid to compromise the privacy of the data owner and the reputation of the publishing institution.

Solutions existing in literature to deal with the modeling and the enforcement of privacy constraints take typically the point of view of the database administrator or the report designer. However, there is no easy way for developers to collect from the owners of the data and the data subjects indications on which data should be considered sensitive and should be protected.

Our idea is to let the privacy expert to say when a privacy violation may occur by showing her sample data. The sample data are selected taking into account the size of the sample a user can manage and the distribution of the data to be analyzed. The goal is to minimize the number of cases the user has to consider (some cases will be redundant or could be inferred from what has already been said) and to show a sample that respect some

constraints (e.g. the size of the screen).
This requires to answer the following question: once the privacy expert has reported all the cases in which a privacy violation occurs, how can we express in a concise form the privacy constraints inferred? In addition, as we gather notifications of privacy violations from the user in an iterative way, we need to find the best way to show the sample data according to a certain order which facilitates the work of the user and which converges to the final set of constraints as quickly as possible and without loosing any privacy constraints.

1.3 Structure of the Thesis

The thesis is divided in the following chapters:
Chapter 2 surveys related works in the research area connected to this thesis and in particular on the design of EHR, on the available architectural solutions for the design of data and application integration systems and on the management of sensitive information with Privacy-Enhancing Technologies. From this analysis we saw the lack of a comprehensive solution covering all our research and operative goals.
Chapter 3 presents an architectural solution for an EHR in which privacy is first class citizen. It presents an application integration solution balancing the needs of visibility achieved with an event-based system with the desiderata of control and protection required in an healthcare scenario.
Chapter 4 provides a theoretical analysis of the privacy constraints definition problem. It presents an elicitation approach based on sampling for the generation of concise privacy constraints minimizing the effort of the user. The constraints produced are directly enforceable in a relational database. An analysis of the effectiveness of the approach on test data is also provided.
Chapter 5 presents the application of the architectural solution of EHR proposed in the previous chapters to real scenarios. We highlight the approach used to analyse the current systems and the effort required to adapt the research solution into the specificities and complexity of real systems. We present the approach used to interact with the user to collect the system requirements in response also to the constraints imposed by the privacy regulations. Finally we test the applicability of our solutions in a particular challenging environment to realize an EHR in Mozambique.
Chapter 6 gives an overview of the key contributions of the thesis and presents the future research directions.

1.3.1 List of Publications

The results of this thesis and part of its content have been published in the paper listed below. The activity related to the CSS (Cartella Socio Sanitaria) project described in this thesis was supported in part by the Autonomous Province of Trento and FBK-Bruno Kessler Foundation and as a joint work of GPI [2] and the Department of Information Engineering and Computer Science (DISI[3]) of the University of Trento.

1. Annamaria Chiasera, Fabio Casati, Florian Daniel and Yannis Velegrakis,*Engineering Privacy Requirements in Business Intelligence Applications*, Proceedings of the 5th VLDB workshop on Secure Data Management, SDM '08, Auckland, New Zealand, pp. 219–228, Springer-Verlag

2. Manuela Corradi, Annamaria Chiasera, Giampaolo Armellin, Jovan Stevovic, *Understanding how people work: experiences in improving healthcare practices in Italy*, Workshop on Coordination, Collaboration and Ad-hoc Processes (COCOA'10), HP Labs, Palo Alto, CA.

3. Giampaolo Armellin, Leandro Paolo Bogoni, Annamaria Chiasera, Tefo James Toai, Gianpaolo Zanella, *Enabling Business Intelligence Functions over Loosely Coupled Environment*, 2nd International ICST Conference on e-Infrastructure and e-Services for Developing Countries (Africomm'10), Cape Town, South Africa

4. Giampaolo Armellin, Dario Betti, Fabio Casati, Annamaria Chiasera, Gloria Martinez, Jovan Stevovic, *Privacy Preserving Event Driven Integration for Interoperating Social and Health Systems*, 7th VLDB Workshop on Secure Data Management, SDM 2010, Lecture Notes in Computer Science, Volume 6358/2010, pp. 54-69

5. Giampaolo Armellin, Dario Betti, Fabio Casati, Annamaria Chiasera, Gloria Martinez, Jovan Stevovic, Tefo Toai, *Event-Driven Privacy Aware Infrastructure for Social and Health Systems Interoperability: CSS Platform*, 8th International Conference on Service Oriented Computing, ICSOC 2010, San Francisco, CA, USA, pp. 708-710

6. Giampaolo Armellin, Dario Betti, Annamaria Chiasera,*Il referto (ri)trovato. Appunti per l'architettura dell'informazione di un Fascicolo Sanitario Elettronico*, IV Summit Italiano Architettura dell'Informazione, Pisa, Italy, 2010

[2] http://www.gpi.it/

[3] http://disi.unitn.it/

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

8. Giampaolo Armellin, Annamaria Chiasera, Ivan Jureta, Alberto Siena, Angelo Susi, Establishing information system compliance: An argumentation-based framework, Fifth IEEE International Conference on Research Challenges in Information Science, Guadeloupe - French West Indies, France, 2011, pp. 1–9,

9. Annamaria Chiasera, Tefo James Toai, Leandro Paolo Bogoni, Giampaolo Armellin, Juan Jos Jara, *Federated EHR: how to improve data quality maintaining privacy*, IST-Africa, Gabarone, Botswana, 2011

10. Giampaolo Armellin, Annamaria Chiasera, Ganna Frankova, Liliana Pasquale, Francesco Torelli, Gabriele Zacco, *The eGovernment Use Case Scenario*, Book Chapter: Service Level Agreements for Cloud Computing, Wieder, P.; Butler, J.M.; Theilmann, W.; Yahyapour, R. (Eds.), Springer New York, 2011

11. Joe Butler, Juan Lambea, Michael Nolan, Wolfgang Theilmann, Francesco Torelli, Ramin Yahyapour, Annamaria Chiasera and Marco Pistore, *SLAs Empowering Services in the Future Internet*, Book Chapter: The Future Internet Lecture Notes in Computer Science, 2011, Volume 6656/2011, 327-338

Chapter 2

State of the Art

The design of privacy compliant interoperability infrastructures in healthcare touches many research fields to properly answer to stakeholders' needs and to legal, organizational and technological constraints.

This section analyses the solutions in literature that can be exploited in devising a privacy-aware Electronic Health Record (EHR) system. They are divided in the areas: Electronic Health Record, an overview of emerging health information technologies, their main functionalities, benefits and experiences from existing implementations; data and application integration patterns that can be used in the design of an EHR solution; approaches to capture privacy requirements from regulations; compliance and provenance management techniques, to monitor the behavior of the system and to support the auditors in compliance checking using provenance as a way to grant visibility of the data management flow; modeling and specification of privacy policies, to express, refine and manage privacy requirements; access control mechanisms, to implement purpose based data management restrictions; privacy preserving in databases and information management, to analyze and integrate data still preserving the privacy of their data sources and data providers.

2.1 Electronic Health Record

Recently, the interest in eHealth systems is increasing [85, 79] as a way to minimize costs induced by a reduction of patient's stay in hospital, an optimization of the work of medical staff in sharing information and an improvement of care quality (e.g. improved control of adverse drug events [107]). Healthcare providers are gradually moving from paper-based medical records to Electronic Medical Record (EMR) and to even more evolved eHealth systems to maintain patient's data.

The European Commission [66] defines an EMR and its evolutions as follows:

- *Electronic Medical Record (EMR): the electronic record of an individual in a physi-*

cians office or clinic, which is typically in one setting and is ***provider-centric****;*

- *Electronic Patient Record (EPR): the electronic record of an individual in a hospital or health care facility, which is typically in one "organisation" and is* ***facility-centric****;*
- *Electronic Health Record (EHR): the longitudinal electronic record of an individual that contains or* ***virtually interlinks*** *data in multiple EMRs and EPRs, which is to be shared and/or interoperable across healthcare settings (****inter-institutional****) and is* ***patient-centric****.*

An EHR allows to share patient's information in form of EMR among different institutions. Recently the trend is toward the adoption of Personal Health Record (PHR) [148] enriching EHR with information provided by the patients themselves.
This poses even more stringent privacy issues as also problems in certifying the quality of the data. According to the DataLossDB [1] statistics 20% of the violations of Personally Identifiable Information (PII) in United States occurs in the medical sector.
In this thesis we focus on EHR as it is a prerequisite to build more evolved eHealth systems and particularly to create a national and international level Health Information Exchange (HIE) infrastructure.
The Institute of Medicine identifies the following functionalities of an EHR [134, 36]:

- Health information and data: the capability of construct, maintain and evolve a comprehensive profile of the patient core information (e.g. a patient summary) with allergies and diagnosis, previous laboratory tests and medication list;
- Results management: enable the sharing of medical results in electronic form for a prompt and quicker consultation among caregivers of different institutions;
- Order entry/management: allows the informatization and standardization of medical procedures and instructions for the medical staff;
- Decision support: helps the caregivers to evaluate the effectiveness of medical procedures and treatments, facilitates the diagnosis process and identifies possible risks for the patients;
- Electronic communication and connectivity: improves the communication among the medical team members belonging to the same division and also to different care providers;

[1] http://www.datalossdb.org/

- Patient support: facilitate the patient empowerment in managing his healthcare data autonomously and the continuity of care with treatments performed directly at home improving the efficiency of the care and of preventive protocols;
- Administrative processes: relieves the medical staff from purely administrative task like invoice management, prescription and admissions management and scheduling;
- Reporting and population health management: provides a comprehensive set of information to monitor Key Performance Indicators (KPI) on the quality and performances of healthcare processes distributed in many institutions and the trends in patient's needs.

Indeed the use of EHR to exchange EMRs (Electronic Medical Records) has many advantages: it allows to exchange quicker patient's information, it is more reliable than paper records that can be lost or deteriorate, it improves the accessibility to historical medical data of chronic patients, finally it facilitate the exchange of EHRs with other hospitals and avoids in this way expensive duplicate tests [117].
Full benefits of an EHR are achieved if all the hospitals involved in patient's care adopt it to share information. As shown in [117], privacy regulations [36] may be an obstacle to the adoption of an EHR solution for two main reasons: on one side patients are reluctant to share information and they should be reassured with the adoption of suitable privacy protection mechanisms, on the other side privacy protection makes the exchange of information more expensive increasing the effort to adopt EHR solutions. This motivates us in devising an EHR solution capable to protect patients' privacy minimizing the effort for its adoption.

The American Institute of Certified Public Accountants (AICPA[2]) and the Canadian Institute of Chartered Accountants (CICA) define privacy in the Generally Accepted Privacy Principles (GAPP [12]) as "*the rights and obligations of individuals and organizations with respect to the collection, use, retention, disclosure and disposal of personal information.*" This definition covers all the aspects of the data life-cycle and in particular [115]: who owns or is responsible of the data generated by a data source; how the data owner can lawfully use the data and release them to an external third party entity; how to transfer data to another party under the same or different jurisdiction of the data owner and control data access according to specific roles and purposes of use; how to transform the data to produce statistical analysis without exposing the identity of an individual; how to store and archive the data for a long time guaranteeing access only to the authorized information; how to destroy the data when the retention period expires. In this thesis we do not deal with the data transformation and data retention problems but we focus on

[2] http://www.aicpa.org/InterestAreas/InformationTechnology/Resources/Privacy

the other issues listed above according to the Italian privacy regulations at national and regional level [88, 57, 25, 26, 121, 120, 6] which impact on the technological solutions adopted to develop an health information system.
One of the first issue privacy regulations are dealing with is how to guarantee access to sensitive information only to the data subjects the data refers to and to the users entitled by law, or by the data subjects themselves, to use such data (e.g. caregivers, relatives or governing bodies). In some countries like in Netherland [61] and in some Italian regions (e.g. Lombardia[3], Friuli Venezia Giulia[4] and in the near future also Trentino[5]) patients and caregivers are identified to the EHR by means of a smartcard. Smartcards are usually combined with a password or PIN and a certificate used to encrypt the information stored in a microchip realizing a strong authentication mechanisms. The smartcards released at national level allow to identify and authenticate the user at national level, to sign to PHR systems and certificate the authenticity of documents.

Alhaqbani in his thesis [13] highlights the importance of guaranteeing to patients full control on which health records can be accessed by whom. In our EHR system we delegate the definition of the privacy policies to the data collectors (hospitals, nursing homes and municipalities) as they already got from the patient the consent and the set of information she allows to share. Our focus is on guaranteeing that the data collector can enforce the privacy policies on the collected data. However, this assumption will be no longer valid when the patients themselves will enter information in the ehealth system by means of a PHR. In this case, the user should be able to define his own privacy policies and new mechanisms to interact with the patients and to share the data among the caregivers should be devised.

The management of user's roles and authentication in an healthcare scenario could be very complex. In the eHealth system of an hospital patients are usually associated to a department (e.g. medicine, orthopedic unit) and inside this to a medical branch. Doctors belonging only to the department and branch in which the patient is hospitalized can access her data. Nurses, instead, are not associated to specific branches but they need to work on all the patients in the department and they should be able to see the data of all the department regardless the branch.
The access rights are not static but they may change and there are exceptions to manage. For example, doctors working in the night and during week-ends should work on all the department even if they belong to a different department and branch in the normal working days. This makes the roles dynamic and hierarchical. Recently some work has been done in trying to represent context-dependent roles [73, 74] but they are not

[3] http://www.crs.lombardia.it/

[4] http://cartaservizi.regione.fvg.it/CrsCentralService/

[5] http://www.cartaservizi.provincia.tn.it/documenti/pagina2.html

yet capable to fully model a complex EHR scenario with different institutions involved. What happens in practice is that the administrative and IT departments of the hospital maintains an LDAP (Lightweight Directory Access Protocol) directory with the users to manage the authentication procedure but not user's roles. Roles are managed separately from the LDAP system using authentication systems like CAS [6] (Central Authentication Service) or Shibboleth [7] using also a workflow engine to manage the dynamicity in the roles. In fact, roles could be very complex and change dynamically based on what the employee is doing at the moment.

In literature there are many standards for managing data representation and data exchange in EHR. An example is OpenEHR[8] [145] that is quite generic and defines template and archetype for the design of EHR systems and for the exchange of health data.
IHE, Integrating the Healthcare Enterprise consortium [87] proposes a solution to create and manage EHR based on Cross-Enterprise Document Sharing (XDS). This solution is based on a central Document Registry (based on ebXML[9] technology) with meta-data to search and retrieve documents from Document Repository at the source systems. The Canada Health Infoway system [39], the AORTA Dutch national infrastructure [128] and NHS Connecting for Health in UK [124] are based on the IHE interoperability solution. A similar architecture is proposed in Italy with IBSE [150] for the interoperability of healthcare systems by means of distributed registries for the sharing of clinical data regardless where they are produced (e.g. by institutions with different administrative domains at different regions). These specifications together with the ICAR project [49] funded by the CNIPA (Centro Nazionale per l'Informatica nella Pubblica Amministrazione, now DigitPA - Ente nazionale per la digitalizzazione della pubblica amministrazione) aim at defining a public system of cooperation (SPCoop specification) for the operative cooperation of public administrations. Our solution extends SPCoop specification with a more fine grained control on sensitive data and a simplified approach to manage the definition of the contracts among data producers and consumers regarding privacy control of sensitive data. In particular, the privacy constraints are not included in the contract signed between a producer and consumer but are managed with detailed policies that further specify the run-time behaviour of the participants in sharing and using data. This relieves us to create and manage specific contracts among pairs of data producer and consumer for each type of data and purpose of use in a P2P style as data access control is managed at the application level.
The experience in Netherlands with the basic cooperation infrastructure AORTA and the

[6] http://www.jasig.org/cas
[7] http://shibboleth.internet2.edu/
[8] http://www.openehr.org
[9] http://www.ebxml.org/

work on realizing an EHR at national level [127] confirms that developing a nation-wide EHR with a central repository is not feasible. Instead, a more conservative approach should be employed where data remains at the information systems of the care providers so that they can continue working with their own specific information systems to request additional data when needed. This is also the approach followed in our EHR solution. Recently there is also the willingness to integrate national-wide healthcare systems to guarantee continuity of care to patients moving around Europe. The epSOS[10](European Patients, Smart Open Services) project tries to standardize common healthcare services like the drug prescription and to define a minimal set of information describing the health status of the patient (*Patient Summary*). The Patient Summary allows healthcare professional to know core information on the patient (e.g. allergies, recent surgeries, medical problems) all over Europe. epSOS uses the IHE specification to define the structure of the documents in such a way that title and header of documents are uniform among different systems and countries and the single fields in the documents can be accessed directly.

The work in [69] by Eze et al. is very similar to our idea of EHR. They analyse typical interoperability solutions based on ERP and SOA showing these are not suitable for eHealth processes. ERP Web portal systems introduce duplication and they force users to search for the data on their own. SOA (Service Oriented Architecture) frameworks avoids duplication but imposes a synchronous interaction mode based on request/response point-to-point invocations. In contrast, eHealth processes have multiple parties concurring in their execution with a point-to-many interaction style over asynchronous and long-running processes. Eze et al. propose an event-driven data integration infrastructure for the palliative care unit at the local health authority in Ottawa with a message broker managing the subscription and delivery of event topics related to the clinical status of patients. The architecture is interesting for our work under many aspects: it uses an event-driven architecture like in our case; it allows to control the way events are routed to the subscribers and to filter the data content with XACML [118] similar to what we do with our privacy policies [18]; it allows to monitor and analyse the inter-department business processes by means of a web portal subscribed to all the events generated, analogously to what our event-feed data warehouse does. The solution is interesting but it cannot be applied as-is in our system since it maintains also sensitive information in the notification messages. Even if they are persisted in the service broker only for the sake of reliable notification they may still contain sensitive data. The solution is perfectly suitable for integrating healthcare systems inside the same institution but is not applicable to integrate different organizations with different privacy requirements and without a shared trusted party (like the web portal) to store potentially sensitive data.

[10] http://www.epsos.eu

In [112] is presented Indivo, a personally controlled health records PCHR (named PING in a previous work [142]) which allows patients to manage and annotate electronic copies (XML documents) about their clinical history, to share them with caregivers from different points of care or with other people (relatives, friends, school and research staff). Data sources can share documents by means of a subscription agent which deals with the synchronization among the data source and the document repository and with translating the documents from the source format to a PCHR-specific format. We used a similar solution also in our EHR system by providing a wrapper module to be applied to the data producers acting as an adapter between the legacy system and the EHR system. The main difference of Indivo from our approach is that Indivo is basically a point-to-point interoperability platform acting between the sources and a centralize data repository while in our case we applied a loosely coupled integration with a message broker distributing the updates of patient's data to the interested parties. This makes the Indivo approach not applicable in our scenario as it duplicates the data even if they are stored in an encrypted data store.

Another PHR based on Indivo is MyOSCAR[11], an electronic medical record which allows patients to control and upload data about their health status on their own, and decide who can see what.

Malamateniou and Vassilacopoulos [111] propose a virtual patient records (VPR) with workflows running among different healthcare providers to exchange patient's data in form of XML documents. Authorization policies are specified to control "*which roles (e.g. physician) are allowed to perform what operations on what data objects and under what conditions*" [111]. Authorizations are defined on the DTD of the XML documents and are used to enforce access control at a global level, for all the healthcare organizations involved, or at a local level inside a single healthcare organization. The idea to integrate the operative work of healthcare organizations by means of a workflow management system is possible also in our EHR as the interoperability infrastructure is equipped with an ESB (Enterprise Service Bus). An application of this approach could be the execution of operations in response to the occurrence of certain events: for example when the record of a patient is modified, a notification message is sent to the family doctor and to the relatives.

Another project worth to mention is the OpenMRS[12] project aiming at providing an EHR for developing countries. It is realized as a client-server application with patient's data stored in a central database exposed by APIs. A web application is also provided to administer and access the data with common functionalities (e.g. search for a patient,

[11] http://myoscar.org/myoscar/

[12] http://openmrs.org/

edit the health profile). This solution is not applicable in our scenario due to the privacy regulations that will not allow the construction of a centralized database. But its approach is similar to what we used to adapt the EHR system for developing countries: a modular architecture with simple data entry and export functionalities.

In the medical domain there are various standards for representing data and documents like for example the HL7 Clinical Document Architecture [64]. Another important standard to mention is DICOM [122], Digital Imaging and Communications in Medicine, for the interoperability of digital medical images and to realize picture archiving and communication systems (PACS) [100]. In our implementation of EHR we used plain XML documents to exchange data to further limit the complexity of the infrastructure for the data sources joining the system. However, the architecture is fully compatible with more advanced standards for the medical domain like HL7.

2.2 Data and Application Integration Patterns for Healthcare

Business Intelligence technologies are being increasingly used by companies and institutions for detecting problems and inefficiencies in the execution of their business operations and for identifying opportunities for improvement. The reason for the recent success of BI solutions and their fast adoption[13] are due to both improvements in BI technology and its key ingredients (ETL, warehousing, reporting, and mining) and in the increased availability of digital information that makes business operation analysis feasible.

Interoperability solutions commonly used for medium and large integration projects that are based on data warehousing (DWH), Enterprise Information Integration (EII) or Service Oriented Architecture (SOA) are not applicable as-is to our scenario [43]. Let's briefly analyze the pros and cons of the different approaches.

In a *Data WareHouse* (DWH) data integration is achieved by means of periodic-snapshots of detailed and aggregated information flowing from the sources to the data warehouse. A data warehouse is basically an integrated database capable to maintain a huge amount of data covering a wide period of time that allows to create a complete history of the information. The internal data structures are typically designed with a *star schema* structure [98] that is optimized for reporting and for the resolution of complex queries.

As shown in Figure 2.1a the bridge between the sources and the data warehouse is represented by the ETL (Extract, Transform, Load) system aiming at:

- **Extract**: it deals with the extraction of data from the sources according to different extraction modality depending on the particular type of source: direct access to the

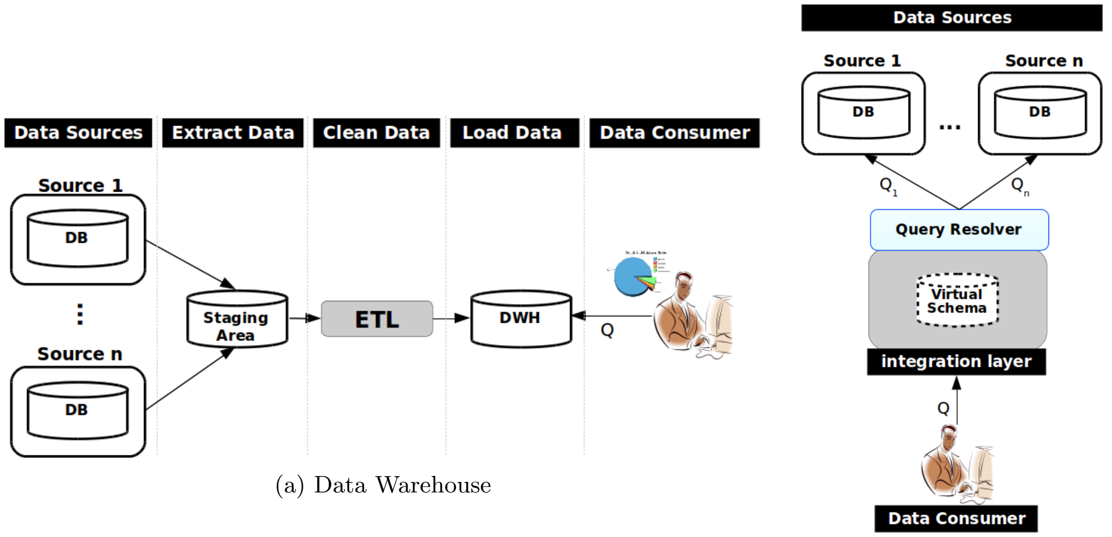
[13] See Gartner, http://www.gartner.com/it/page.jsp?id=501189

DB, periodic data pump by email or FPT, creation of specific infrastructures at the data source dedicated to the data extraction. The extracted data ends up in the so called “staging area”, an area that is not accessible by the final user of the DWH in which the data are further transformed with cleaning and standardizations steps to fit into the DWH data structure to guarantee suitable level of quality.

- **Transform and Load**: data extracted from the sources is checked to identify errors, missing data, mismatches and duplicates, that are corrected or notified to the DWH administration for a manual resolution. The staging area represents a central place in which all the information necessary to perform the consistency and correctness check of the data are available (e.g. mapping tables internal or external to the DWH and logging tables). Depending on the number and type of sources the data cleaning and transformation step can be more or less complex and it is often necessary to resolve conflicts due to different representation types (e.g. the same data could be labelled with different names). Another problem is due to the synchronization of the data coming from different sources at different instant of times and arrival rate. Once the data transformation phase is completed, the data can be loaded in the data warehouse (loading phase). The staging area allows to collect and clean the data in an incremental way and even asynchronously. This makes the whole system more robust to unavailability of the single sources.

Adopting a DWH to realize an EHR is not feasible as: (i) it requires to duplicate data and this is not allowed by the privacy regulations for EHR [88]; (ii) data flows only from the sources to the DWH while the reverse is difficult and makes hard to get a full interoperability among the systems; (iii) the data synchronization occurs periodically while in the healthcare domain is important to access to the most recent data in a timely way.

Enterprise Information Integration (EII) defines a virtual schema as a logical view on the sources [80]. As showed in Figure 2.1b, the data remains stored at the sources and the EII integration layer encapsulates the mapping and transformations of the information from the virtual integrated schema to the single sources. The integration layer exposes to the final user only the virtual schema on which to perform inquiries while the heterogeneity of the sources is hidden behind. Each query defined on the virtual integrated schema is decomposed by the integration layer into sub-queries that are forwarded to the single sources. Each source will send back to the integration layer of the EII the answer to the sub-query that will be further processed and integrated (by a query resolver module) to be returned to the user as a reply to the initial query.



(a) Data Warehouse

(b) Enterprise Information Integration

Figure 2.1: Data Integration patterns.

The use of a unique unified schema to query allows to act independently from the actual location and format of the data and from the protocols and security constraints in use at the sources. Information is retrieved at query time directly from the sources in the more up-to-date status. This unfortunately makes the whole system strongly dependent on the availability of the sources: if a source is unavailable or is not working properly the results obtained could be incomplete or erroneous . Furthermore, this query resolution mechanism may hinder and slow down the operational efficiency of the sources as it requires them some effort to retrieve and process the required data. This induces a degradation of the response time especially for those sources with poor computational capability.

Notice also that, in contrast to a DWH solution, there is no staging area available to perform harmonization and cleaning procedures and the partial results coming from the sources are combined on-the-fly. This makes more difficult in EII to guarantee data quality as there is no enough "room" to detect and correct inconsistencies in the data coming from the different sources within suitable response time.

The EII approach is recommended when it is possible to have a direct access to the sources, for example within an intranet or when the sources belong to the same administrative domain.

In contrast, an EII solution requires too strong requirements on the data sources when they are heterogeneous, belonging to different administrative domains or with unreliable access. Furthermore, it strongly depends on the performances and availability of the data sources: a wrong or incomplete answer due to unavailability or inconsistency among the sources is not acceptable in healthcare.

The use of Web Services (WS) and *Service Oriented Applications* (SOA) for data integration [15] is based on exposing resources and functionalities of the applications as services. SOA uses two key concepts: the concept of **standardization**, specifically at the level of interaction protocol; the concept of **loose coupling**, that is, services that are not written with a particular client in mind but that are designed and maintained to be useful to a large number of clients. SOA solutions are particularly suited for the integration of applications within a single organization but also among different organizations. Recently SOA techniques have been reinterpreted in the context of data integration: data resources have been exposed as services and the access to the data is performed by means of services invocation. There are three main modalities in which services are used for this purpose: (1) as an *ETL adapter*, providing to the ETL tool a uniform mechanism to access to the sources; (2) as a *query adapter*, providing an interface for an EII application or a DBMS to perform SQL-like query on the service; (3) as a *monitor*, that observes the source and emits events informing on the data changes. Such a notification can be in batch or real-time mode (i.e. each change generates an event). The last modality leads to the Event-Driven SOA Architecture (EDA) and is the one we decided to use in our EHR solution as shown in Section 3.1. In our EHR scenario there are many actors impersonating dynamic roles and increasing in number as more institutions tend to join the system. SOA pattern is well suited for point-to-point synchronous interactions but in this intricate scenario it becomes soon unmanageable.

Event-Driven SOA Architecture (EDA) mitigates the problems highlighted for the pure DWH and EII solutions with a solution that is extremely loosely coupled and distributed [116]. In event-driven systems data producers exchange data with data consumers by means of asynchronous events and the mediation of a common event manager (a messaging system [86]) as shown in Figure 2.2 (the picture is taken from [68]). As said in [67] an event-driven architecture allows the *personalized delivery of information* that is to "*deliver the right information to the right consumer at the right granularity at the right time*" and brings the following advantages (see [67, 86] for a complete list):

- *Abstraction*: the operations for generating and processing the events can be kept separated from the application logic and they are typically isolated into adapters modules [15]. In this way the adapters can evolve without impacting on the producing and consuming applications. This has also the benefit of abstracting from the complexity of the application logic and of the data structures of the systems to be integrated.

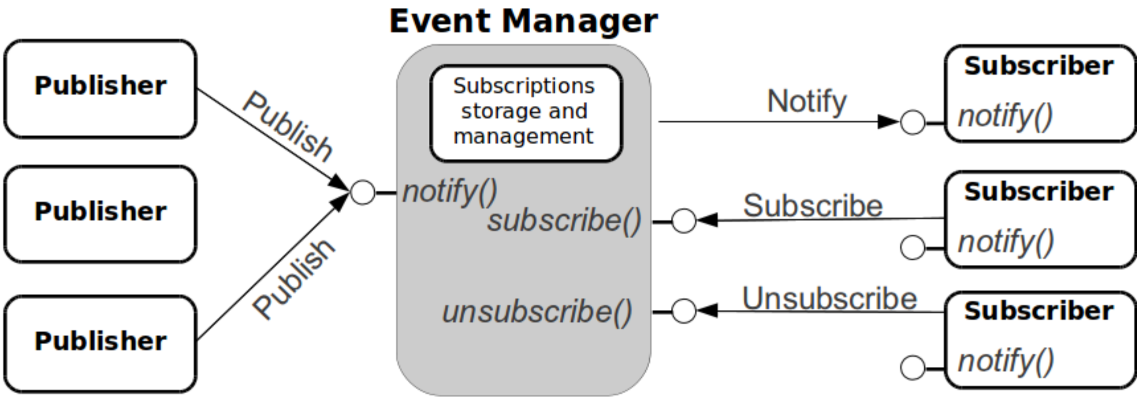


Figure 2.2: Example of basic publish/subscribe system from [68].

- *Decoupling*: there is no need for the data producers and consumers to be aware of each other as both interacts only with the event manager. In this way an event generated by one producer can be broadcasted to many different consumer applications.
- *Asynchronicity*: it is not necessary for the data producers and consumers to be up and ready at the same time and event processing can be performed asynchronously. This is particularly useful when an application is temporarily unavailable or events are produced at irregular rates and the processing of event picks can be deferred when the consumer is less busy (e.g. during the night).
- *Data timeliness*: by sharing small chunks of data frequently is easier to keep producers and consumers in sync minimizing the time interval in which they are not aligned. This makes the interaction between consumer and producer more natural and similar to what happens in the real world (e.g. admission and discharge of a patient can be represented by two events). The data consumers do not need to query the data producers to get new data as it happens in pull mode interaction. But consumers are notified by the data producers when new data arrives with a push mode interaction. This allows the data consumer to continue its work without losing any key update.

Note how EDA is not an alternative to SOA but in many cases they are used together. The use of SOA combined with EDA allows to share information efficiently, with minimal effort and in a timely manner [69]. Recently this combined approach is indicated with the term *event-driven SOA* which takes advantage of the benefits of the two approaches [67]. Business Activity Monitoring (BAM [84]) is part of the Operational Intelligence analysis and leverages the capability of real-time continuous analysis by capturing events to discover and notify anomalies.

In our EHR solution we adopted a mixed EDA-SOA driven approach [116] in which involved entities exchange data through WS invocation and the data processor implements the Publish/Subscribe [68] functionalities. Our approach is similar to [119] with events used to transfer information by pub/sub but our case is more general as: it is not limited to mobile devices acting as data producers but it can work with any source system; it maintains a centralized index of the events with only meta-data on the real event occurrence to search for them even after a long time elapsed from their publication to the subscribers; the content of the event is maintained at the producers for legal reasons to agree to privacy regulations (Section 3.1 explain our solution more in details).

The differences of our solution with similar systems based on EDA and in general with the messaging systems are due to the peculiarities of the EHR scenario in which events should remain available after the notification at any time and for the whole lifetime of the patient among heterogeneous systems. This requires to persist the notified events so that they can be indexed and queried even a long time after the notification. The Integrating the Healthcare Enterprise - IHE consortium [149] proposes a solution based on a central registry of searchable meta-data linked to the data generated by producers. A Cross-Enterprise Document Sharing (XDS) cooperation architecture allows the document sharing (typically documents with simple text, formatted text in HL7 CDA Release 1, DICOM images) among the federated document repositories at the healthcare delivery organizations using an ebXML Registry. The solution of the IHE consortium is applied to many eHealth projects around the world: NHS Connecting for Health[14], Canada Health Infoway[15], AORTA[16] (the Dutch national infrastructure).

Various implementations of registries like UDDI, ebXML (Electronic Business using eXtensible Markup Language, e-business XML), XML/EDI [154] are currently available. The most appropriate in terms of flexibility, interoperability and also the more widely used for the Health domain [63, 62] is ebXML by OASIS [129]. The ebXML registry implementation is also adopted in our EHR.

The problem of integrating information from different and distributed sources is well studied in literature [102]. Jeff Ullman [89] identifies three main integration architectures: *federation*, in which data sources talks each other in a peer-to-peer fashion; *warehouse*, in which data at the sources are transformed according to a global schema and copied in a central DB (a data warehouse); *mediator* (or virtual warehouse), in which a mediator layer translates user's query into a sequence of queries executable at the sources and the data remains at the sources.

[14] http://www.connectingforhealth.nhs.uk/

[15] https://www.infoway-inforoute.ca/

[16] http://www.ringholm.com/docs/00980_en.htm

The first approach is also named Peer Data-Management by Alon Halevy [14]: in this approach, each peer can provide a view of its data or behave as a mediator for the other peers propagating the query to other peers if it is not capable to answer on its own. The answer to a query is returned directly to the requester and so there is no privacy issues. However, we did not apply this approach for the EHR proposed in this thesis, because in this case there is a partner which can act as central data processor to trace all the access requests. The tracing is necessary to keep track of who access what for auditing purposes, and in a peer data-management it is not easily doable as answers to queries can be provided by any source in the system.

An information integration system using a mediator approach is composed of a wrapper module at each source and a mediator layer. The wrapper module is in charge of preparing and sending the data from the data source to the mediator. The mediator encapsulates all the logic to integrate the data from the different data sources. It retrieves, composes and integrates the data obtained from them and provides the integrated data to the consumers in such a way that the consumer is not aware of the differences at the data source level.

In our EHR solution we combined the mediator approach with a publish-subscribe approach, as it allows to inform multiple consumers with notification messages signalling a data update is ready. In this architecture the sources can behave both as producers and consumers. The mediator plays the role of coordinator of the publish/subscribe system to assure that when an event arrives it is delivered only to the subscribers. The advantages of the publish/subscribe approach for the consumers are: (i) they get immediately useful information from the notification content (e.g. the notification of the start of a clinical examination) without querying further the sources; (ii) they can go ahead with their internal operations, and query the sources asynchronously only when the desired information is ready and the related notification message has been received.

We also have a DWH in the architecture but its role is to keep a log of all the data transferred in the system. In contrast to standard DWH systems, when a request from a customer arrives to our system the data are retrieved from the sources instead that from the DWH. In this regard the DWH behaves like a data consumer.

The implementation of the communication between the producers/consumers and the mediator is done through an ESB (Enterprise Service Bus). So, from now on, whenever we refer to the term bus we mean the term ESB.

2.3 Privacy-Enhancing Technologies

In the previous sections is presented what an EHR is, how it is typically realized and which technologies can be used for its design and development. This section presents techniques that can be used to guarantee the EHR complies to the privacy regulations both in the design and in the behaviour at run-time.

Privacy Requirements Gathering

The problem of defining and refining privacy requirements is similar to more general software engineering problems and specifically requirements analysis, refinement and management. Privacy requirements should be elicited, analyzed, defined and verified that these are actually met in the system applying a suitable development process and set of best practices. Customers are often imprecise in explaining their expectations from the system (or they simply don't know what they want in the beginning) so that requirements tend to evolve and become clear only after repeated interactions with the developers. All these issues are well known and also well studied in software engineering [143]. Research in this area has lead to modeling languages (as UML, Unified Modeling Language), software development processes and agile methodologies (as the RUP, Rational Unified Process, and the SCRUM approach), and tools to aid the software engineer in performing the requirement engineering phase in a way as accurate and as painless as possible to minimize the mismatch between customer's expectations and what the system actually provides. Evolutionary development processes based on rapid prototyping and testing of the intermediate implementations of the system are often the best choice when requirements are fuzzy and inclined to change as is the case with privacy requirements.

UML and other modeling languages from the requirement engineering community, e.g. i* [60], may also be used to represent privacy requirements. These languages are expressive enough, but hard to use, and due to the fact that they have been to a large extend ignored by modeling techniques, their integration into a data management solution is not an easy task.

An alternative option is to model privacy requirements in terms of meta-data that accompany the data and controls its access and use. The advantage of this approach is that the metadata can be easily defined and can accompany the data throughout transformations [146]. The meta-data can be part of the data model, typically as data annotations [47, 76]. At a certain extend also in our EHR system privacy constraints are represented as meta-data associated to the database views we want to protect and to the events transmitted among the parties.

Privacy regulations impose constraints on the design of an EHR and also on its run-time behaviour [135, 7, 57, 88]. Such regulations depend on the country and in some cases they may prevent data sharing among different jurisdictions. The U.S.-EU Safe

Harbor framework[17] was introduced to allow US organizations to share citizens' personal data with EU countries in compliance with the EU directives. In [32] are highlighted the requirements an organization processing sensitive information about health should satisfy to be compliant to the HIPAA regulation.

In [132] it is highlighted the importance of understanding which are the privacy and confidentiality policies and rules that are in place to avoid sharing sensitive information with other organizations (either private or public) that is not strictly needed to manage the patient. This is also known as the collection/data minimization principle or minimum necessary disclosure [28]. On the other hand if the system is designed to be too much conservative it results in over-constraining the data and in the end in a system which is unusable for the users.

Furthermore, is important to adopt standards to represent healthcare information to allow the data exchange between EHRs at different organizations and jurisdictions and with the public healthcare organizations.

In [99] is proposed a tool to automatize the analysis of regulations written in natural language with legal terminology. The tool uses semantic annotations to identify rights and obligations in regulations like HIPAA. The automatic analysis of regulations to derive system requirements, or at least to model the obligations in such rules, is not simple as law typically contains a certain degree of uncertainty and ambiguity and manual intervention is needed [22].

Lam et al. [101] proposes another systematic approach to formalize regulations as Datalog logical rules. In this way it takes advantage of the possibility to verify the satisfiability of the rules to check compliance and identify conflicts. The effort of formalizing the natural language regulations into Datalog rules could be quite high and limit the applicability of the approach. It is also difficult to use Datalog rules to interact with the medical personnel as instead is suggested in the paper to train the users on the importance of satisfying regulations.

In [40] is analysed the problem of managing privacy obligations imposed by privacy laws (e.g. HIPAA [7], COPPA [71], GLB [70], Data Protection Act [136]) and guidelines (e.g. OECD guidelines [131]) in enterprises and organizations. It proposed an architecture in which data is stored in obfuscated form by means of cryptographic techniques and data events are generated from the data repositories, databases or file systems. Events are monitored and privacy obligations are enforced to avoid the disclosure of sensitive data. The approach is quite general and do not propose an implementation of the high level architecture. It emphasises that the deploy of a privacy obligation framework should require minimal impact on the applications and services in

[17] http://www.export.gov/safeharbor/index.asp

use in the companies and organizations. In this regard, the approach is quite similar to our solution as we also enforce privacy policies on events with a solution that minimizes the impact on the data sources [18].

In [105] is proposed a modeling language to express actionable requirements to satisfy policies (like the Safe Harbor) on outsourced data. For example, it allows to express retention policies and to verify data are deleted at the expiration of the retention period. It also provide a run-time environment to enforce the actions as a state machine in a workflow engine.

The approach presented in [95], and further extended in [22], applies an argumentation framework to reason on legal concepts in a systematic way with the users. The approach uses the *Nòmos* notation to link requirements with the legal concepts and an argumentation process to trace the motivations (arguments) leading to a certain requirement, the achievement of a goal and at last the satisfaction of a law. In this thesis we applied the argumentation approach to motivate to the domain experts and privacy auditor the compliance of the EHR design to the privacy regulations. The approach is effective but it requires some effort to be applied on complex system and it is worth to use only on well delimited scenarios and for the most critical legal constraints. Other compliance management techniques are further analysed next.

Compliance and provenance management techniques

It is important to note how verifying the correct enforcement of privacy restrictions can be seen as a special case of compliance checking and should be supported by suitable compliance management techniques. Compliance checking means to verify at which extent a system, and in general an organization, satisfies regulations, best practices and codes of conduct. An example of compliance to financial regulations is the monitoring on financial transactions performed by anti-money laundering activities. With the introduction of legislature and regulatory bodies [140] as Sarbanes-Oxley [4], Basel II [133] and the COSO framework [3] for the financial world and HIPAA [7] (Health Insurance Portability and Accountability Act) for healthcare information, compliance has gained increasing importance from a legal point of view with punishments in case compliance check is not passed.

In Business Process Management systems (BPMSs), techniques have been proposed not only to detect lacks in achieving compliance (detection approach) but also to guide the design of Business Processes (BPs) to satisfy the requirements derived from compliance needs [33] so that compliance is “achieved by design” [140] (preventive approach). Governatori et al [77] provides a logic-based formalism to express contracts and obligations in business processes. Compliance of a BP to the contracts is performed by monitoring the generated events to check they comply with the contract.

Compliance requirements could be rather complex constraining the actions and steps the BP is allowed to do (or has to do) on resources and data exchanged between BP tasks. In [96] are studied the problems of how to verify if the BP is compliant to such complex constraints and also how to auto-generate processes from them.

Agrawal et al. [9] propose a framework to support requirements imposed by the Sarbanes Oxley act as auditing and compliance verification of internal financial activities carried out by a company.

Process monitoring techniques [41] could be valuable means to address compliance needs with the collection and analysis of information on running processes to discover bottlenecks, points of improvements, discrepancies with constraints and expectations usually formalized into SLAs at modeling time [21, 38].

A more privacy centric approach is used in data intensive applications like Hippocratic DB [10] where compliance to HIPAA is achieved by the enforcement of privacy restrictions at query time and the logging of queries to help the auditor in discovering privacy breaches.

Chinosi et al. [46] provides a compliance checking procedure to verify a business process is compliant to P3P (Platform for Privacy Preferences) privacy policies by extending an XML-based BPMN representation (BPeX) with attributes that can match P3P policy elements.

In our EHR system information may be derived by integrating data coming from different sources imposing different privacy restrictions. Provenance (and its synonym lineage) with its capability to capture the origins of data [147] can facilitate privacy and compliance management. It allows to build the tracing metadata required in compliance checking activities to understand the behavior of a system when it transforms the data. Provenance may be applied to improve and control data quality especially when provenance metadata could be queried (e.g. to retrieve the source of uncertain information in a database as proposed in the Trio system [155]). It is used to describe the derivation process of a piece of data so that it could be replicated elsewhere by re-running the recipe given in the provenance data (e.g. in scientific workflow or curated databases) or explored by an auditor to understand how data are managed to comply with particular privacy rules and in general for compliance management.

Data provenance systems could be classified in annotation based and non-annotation based [147].

In the annotation based approach provenance is achieved by means of annotations associated to the data. Annotations evolve with the data to reflect the transformations applied in the data flow from the sources to the target repository. In that way provenance is always available without any further analysis of the original and derived data. However, this approach may require a considerable space occupancy to maintain the annotations and effort to propagate them on the way of the data derivation process.

Non-annotation based approaches as used in [54] by Cui and Widom do not associate annotations to the data but compute provenance at tracing time. The approach in [54] exploits some properties of ETL transformations to obtain the set of tuples in the input data set contributing in the derivation of the output.

In our EHR system we annotate the data with their provenance (producer and time of generation of the events) to treat the data with different access control profiles depending on the producer and also to trace the quality of the data generate by a certain organization.

Modeling and specification of privacy policies

In the area of privacy, the Platform for Privacy Preferences (P3P) [53, 138, 29] project proposed by the World Wide Web Consortium (W3C) allows web sites to define privacy policies in a machine readable way. A user agent could detect mismatches between the web site's privacy policies and the privacy preferences of users defined in APPEL (the W3C's P3P Preference Exchange Language) and refuse data disclosure [17]. Compact version of P3P policies can be produced to include them in the HTTP header [29]. The applicability of P3P is undermined by its lack of a formal semantics so that different user agents may interpret P3P policies differently. In addition, the expressive power of APPEL is limited and it may be difficult and error prone to define even simple privacy preferences [52, 11, 29]. P3P is thought to be used to define a sort of agreement with the user by publishing the privacy policies the web site is going to use. However, the enforcement of such privacy policies is not addressed in P3P. XPref tries to overcome the limitations of APPEL with a preference language using a strict subset of XPath 1.0 [11]. IBM's Enterprise Privacy Authorization Language (EPAL [23]) and the OASIS eXtensible Access Control Markup Language (XACML) are access control languages that allow to express directly-enforceable privacy policies with purpose of access and obligations (e.g. log access operations) [16]. The two languages differ in the functionalities supported (EPAL offers a subset of XACML's functionalities) and in the way conflicts are resolved. In EPAL, rules are evaluated in sequential order until a rule applicable to the request is encountered. The first-applicable ruling approach [126, 30] avoids conflicts but makes the automatic integration of different privacy policies problematic. XACML resolves conflicts among policies and rules in a more flexible way on the base of a combining algorithm.

The eXtensible Access Control Markup Language (XACML, [118]) is mainly used to control access requests to a web service [48]. As pointed out in [113] XACML considers

the objects to protect as XML documents or part of it that fit perfectly in our scenario, but could be a limitation if data in other formats should be protected.
In this theses, XACML is used to enforce privacy policies not only at document-level but also at a more fine-grained level constraining the access to the fields of the exchanged documents (that in the EDA architecture corresponds to events) [18].
In [56] is proposed a fine-grained solution for the definition and enforcement of access restrictions directly on the structure/content of the documents providing XML responses with their Document Type Definition (DTD) associated files. In this thesis we apply the same idea using XML schema (XSD) instead of DTD as it is more suitable for Web Service invocations.
Jin et al. [90, 91] propose an approach to specify access control policies based on the specified purpose and on a categorization of the sensitive parts of an EHR document. They investigate on resolving conflicts and redundancy among policies from different sources integrated in the same virtual schema. In our EHR we do not deal with conflicts among policies from different data sources because in our system we do not provide an integrated schema on which to map the information and so there is no need to guarantee the privacy policies defined on the events are coherent. Our EHR infrastructure deals only with routing of information to the correct consumer for the purposes and with the content allowed by the producer. Each data source acts independently from the other in defining such sharing preferences by specifying their own privacy policies. This means that a piece of information that is considered sensitive for a data source can be freely released by another. However, privacy policy conflicts will be a problem when all the information coming from the different sources are combined to create a unique shared schema like in the data warehouse module of our EHR populated with the events generated by the different data sources. In this regard the strategies proposed by Jin et al. could be useful.

Access Control Mechanisms
Access control mechanisms grant access to data objects and resources in general (e.g. tables, views, reports, files) only to authorized subjects (e.g. users or applications acting on behalf of the users). They can be enforced at the application level (e.g. embedded in a reporting tool) or at the level of the database storing the data. Reporting and business intelligence frameworks rely heavily on Role-Based Access Control (RBAC) [72] mechanisms to restrict access to reports on the base of user roles. In RBAC, users are assigned to hierarchical roles reflecting the activities and functions carried out in the organization and access rights are associated to roles simplifying their administration. When a user changes his role he will “inherit” automatically the privileges of the new role

[34]. Furthermore, role hierarchies allow to realize separation of duties and make sure a role gets the minimal capabilities necessary to carry out its job. In some cases there is the need of a more fine-grained access control so that access restrictions could be defined on the content of the data to be accessed (Content-based Access Control) in addition to the profile of the data consumer [34]. Consider for example a table containing clinical data of patients in a hospital. In this scenario, medical data of a patient could be accessed only by the nurses working at the floor where the patient has been admitted. Views could be used to grant content-based access exposing to the user only the data his role is allowed to see. In principle we can define a view for each kind of access to implement a view-based access control but this solution will not be scalable [139].

In [108] a query rewriting approach named Query Filter (QFilter) is proposed specifically for XML documents. QFilter allows to enforce fine-grained access control rules on the content of XML documents by means of a pre-processing step which re-writes the input XPath query on the base of the access control policies to return only safe data.

Commercial relational databases like Oracle provide automatic mechanisms (Virtual Private Database, VPD) [5] to enforce fine-grained access control at the row level by means of transparent query rewriting. Queries coming from the users are rewritten by the VPD functionality filtering the data on the base of the privacy policies and users profiles [34]. Such a powerful mechanism may lead to subtle errors due to its transparency. As pointed out by [139] a transparent query rewriting mechanism could produce spurious answers since the query issuer is not aware of the rewriting process that may produce partial results computed only on the limited part of data visible to the user.

When row-level access control is not enough, a more fine-grained control could be performed at the field level even if this is not easy to implement. With a field-level access control the same data may have different access control views. This may lead to problems (polyinstantiation) when update operations should be managed [34].

Access control mechanisms achieve confidentiality that is one pre-requisite for privacy. However, privacy requires also to manage data provider consent and data usage constrained to specific purposes [34]. In P-RBAC (Privacy Aware RBAC) [125] the notions of purpose, condition for data usage and obligations have been introduced into the RBAC mechanism to make it privacy aware. P-RBAC extends RBAC [126, 125] with privacy annotations to support purposes, conditions, actions and obligations. P-RBAC [126, 125] allows to define for each data object the intended usage of that data object, i.e. the purpose. In particular, a data access request is authorized if all the conditions in the request related to role, data requested, action performed and purpose matches the permission assignments corresponding to the policy. The main contribution of P-RBAC is the unification of privacy policy definition and enforcement with access control mechanisms and the proposal of conflicts detection procedure (although only between pairs of policies). In this thesis we adopted a similar approach but with the difference that we defined policies regulating the access to single fields inside an XML document.

The languages presented so far are of generic nature and can be used in different contexts. However, due to their generality, they cannot easily express actionable privacy requirements that are directly "testable" and "verifiable". They are neither intuitive enough to be used by a privacy expert to express even simple constraints. Furthermore, they require a translation step to make the privacy constraints enforceable on the data schema.
A different approach is the exploitation of the notion of views. In particular, views are defined to disallow or restrict access to the base tables specifying different permissions and operators in each one. The use of views has the additional advantage that it can combine information that is distributed across different tables, thus defining privacy restrictions on the integrated information would have never been possible by defining restrictions on the individual base tables [34]. Our approach based on samples and constraints specified on them inherits the advantages of the view definition with the difference that it relieves the user from the definition of such views.
Alternatives or complements to the use of views as an access control mechanism include automatic query rewriting techniques, such as those found in commercial databases like Oracle Virtual Private Database (VPD) or in the Hippocratic Database (HDB) [10]. In [55] privacy constraints are expressed as queries that is as views on the database. Enforcement is performed by checking query/view containment. Incoming queries may be completely rejected or rewritten in order to release at least some minimum information. The problem with this approach is that it requires to define the privacy constraints as SQL queries which limits its use only to database administrators that typically are not experts in privacy. In contrast, our approach is thought to interact directly with the privacy expert with no mediation by the database administrator.

Other privacy preserving query answering techniques are based on the perturbation of the output expected from the queries [137]. Perturbation can also be applied to the data in input by adding noise in such a way that the statistical distribution and the patterns of the input data are preserved and the quality of aggregate reports or mined results is not compromised, even if derived from altered data [152]. Cryptographic techniques can be used to scramble the data, again without compromising the possibility of computing aggregates or mining data [152]. These techniques are very useful to perform statistical analysis but cannot be applied when the end-users need access to the exact information as it happens in the healthcare domain.

In our EHR system we applied cryptography to the identifying information of the citizen stored in the DWH and used to correlate the events, specifically *Name*, *Surname* and *Fiscal Code*. This implements the request of the Italian Data Protection Authority [88] to "*prevent the data from being intelligible to unauthorised entities*". However, it does not protect from a curious attacker that can still infer the identity of an individual by collusion attacks as occurred in famous cases like the privacy breaches induced by the release of search queries by American On Line in 2006[18].

Privacy Preserving in databases and information management

A common way to protect privacy used in the statistical database community when detailed private information are published is to manipulate the original data (*sanitization*) so that they become anonymous and can be freely published without compromising privacy [8]. In alternative, sanitization can be applied to the output of the queries leaving the original data as they are.

Many techniques has been developed in Privacy Preserving Data Publishing (PPDP) to deal with the publication of microdata with certain requirements of privacy and possibly minimizing alteration of the original data. Most of the sanitization techniques work on the so called quasi-identifier or pseudo-identifiers fields (e.g. age, zip code, sex) [8]. Quasi-identifiers are attributes that could be linked easily with external publicly available information (e.g. voting registry or census data) to identify with a good estimate sensitive data (e.g. more sensitive fields like a patient's disease) [144] by inferring the real identity of the individual (e.g. name and social security number).

The intuition behind anonymization techniques like k-anonymity [144] or l-diversity [110] is to hide a person in the crowd [109] to avoid linking attacks. Re-identification of individuals by linking attacks is based on the use of quasi-identifiers released together with the sensitive information that can be linked to publicly available data containing the actual identity of individuals.

These anonymization techniques make it hard for an attacker to successfully perform linking attacks by hiding a single individual in a group of similar records to avoid an exact re-identification. This naïve idea is formalized in the concept of *k-anonymity* where individuals are clustered together into groups of size k so that they share the same value for the quasi-identifier fields with other $k-1$ people. Each individual is identical to the others clustered in the same group when viewed restricted to the set of quasi-identifier fields. Intuitively, the bigger the group the harder is to identify a single individual and its sensitive fields [144]. Sweeney [144] proposes two approaches to built a k-anonymous version of a table by means of generalization and suppression repeatedly applied until

[18] http://select.nytimes.com/gst/abstract.html?res=F10612FC345B0C7A8CDDA10894DE404482

the table becomes k-anonymous. Generalization substitutes a value with a more general one. The maximum generalization step corresponds to the suppression of the element (field or tuple).

Anonymization induces a loss of information due to the fact that some details are no longer present in the generalized data. A trade-off should be reached between the level of protection achieved and the loss of information.

Unfortunately, k-anonymity techniques are not completely safe from privacy breaches. If all the records in the group of k identical tuples of a k-anonymous table share the same values for the sensitive data, then the approach is totally ineffective as shown in [110] with the concept of l-diversity. In l-diversity the k-anonymity condition is extended to solve its weakness so that for the tuples with the same values for the quasi-identifier fields the values over the sensitive attributes should be different.

However, also l-diversity may fail in certain cases and more advanced techniques have been proposed like (α, k)-anonymity [156] and t-Closeness [106] which enforce more strict requirements in the data distributions of the released table.

In [158] Xiao and Tao propose to release the quasi-identifiers and the sensitive values into separate tables. The sensitive fields of the tuples in a bucket are mixed together so that there is no more a one-to-one correspondence between the quasi-identifiers (QI) and the sensitive attributes but a single QI can be associated to any of the sensitive attributes.

An interesting approach for releasing safely sensitive data is proposed by De Capitani et al. [59]. They specify *confidentiality constraints* to protect sensitive associations among data (e.g. the publication of the name of patient together with his diagnosis) and *visibility constraints* imposing instead certain data to be published together (e.g. the birth date and zip code of patients should be released or the SSN). The confidentiality constraints are satisfied by releasing the data into fragments in which sensitive data do not show up together neither in a single fragment nor in their join. Fragments are constructed with no attributes in common to inhibit join operations among them and they are split in such a way that both the visibility and confidentiality requirements are satisfied with a minimum number of fragments. The authors reduce the fragmentation problem to a SAT problem with boolean formulas representing the constraints and they propose a resolution algorithm based on a SAT solver.

The confidentiality constraints proposed in this work are similar to the privacy constraints defined in our EHR system as they are both inspired by the idea that releasing certain attributes together may compromise the privacy of an individual. Also our idea of splitting the data into different messages can take advantage of the fragmentation approach to devise a methodology to choose how to split the data in different messages. Currently the only split of attributes which takes into account privacy constraints is

performed among the notification of an events, with the identifying information of a patient, and the details of the events, with the sensitive information. But no other attention nor methodology, apart from the visibility constraints imposed informally by the users on which data to include in the events, are considered in defining the structure of the details and by joining different details is easy to expose sensitive data. The problem with our current approach is that all the notifications and details messages share the same identifying information to be able to correlate the events to a specific patient. This is necessary to guarantee the operative business processes receiving the events at the consumers can correlate them easily and precisely. However, for the sake of the business intelligence tasks could be enough to guarantee that only certain associations among the data is preserved without exposing the real identity of the patients.
De Capitani et al. provide an approach to publish associations (named *loose associations*) among tuples in different fragments by hiding the tuples into groups so that associations are exposed only at the group level. This approach can be applied to create events releasing to a data warehouse module only the data at the sources that can be safely used for the statistical analysis (outsourcing of the business intellicence tasks) or to expose the reporting results without compromising the confidentiality constraints.

Anonymization like k-anonymity [144] or l-diversity [110] can be applied in the cases in which we cannot alter the original information with perturbation or randomization techniques (e.g. results of clinical trials for a doctor) as long as it is acceptable to lose some details (e.g. the exact place of birth or day of birth of a person). However, they are rather expensive from the computational point of view and usually cannot be applied to anonymize all the data in a database or a data warehouse. For that reason is not advisable to apply them to the whole database but just to some tables and for few fields. Specifically in the EHR system developed in this thesis we plan to anonymize just some report extracted from the DWH before they are published (e.g. the distribution of diseases by place, age class and sex).
The applicability of all these techniques depends on the successful identification of the set of quasi-identifiers based both on the capability of the privacy expert in identifying such key fields and in an evaluation of the knowledge an external attacker can exploit.
Our approach can support all these anonymization techniques once the privacy constraints have been defined and helps the user in identifying the set of quasi-identifiers by giving evidence of the effects of such constraints on the views derived from the original database.

The paper by Atzori et al. [123] depicts a scenario where just discovering the presence of a person in a database represents a privacy violation even if the sensitive information remains unknown. The authors present metrics to quantify the risk of guessing the presence of an individual in a database and to mitigate such risk. This solution helps to deal with the common practice used in hospitals of communicating the personal data and the ward of hospitalization of a patient to the information centres to facilitate visitor's access. The Italian Data Protection Code [57] state that the patients are entitled to disable the sharing of this data to the public and in this case it should be guarantee such information is no more publicly available.

In [109] privacy is preserved without sanitization. The authors propose a way to answer queries only if they do not expose any private data, otherwise they are rejected. The problem of the analysis of the safety of the incoming queries is reduced to a conjunctive query containment problem and the paper shows in which cases it could be solved.

Another work achieving privacy without necessarily use sanitization techniques is represented by the Hippocratic Database, HDB [10], a privacy aware database implemented as a middleware layer on top of existing commercial databases like DB2. Privacy restrictions are defined as corporate privacy policies (written in P3P language) and opt-in and opt-out choices given by the data provider to define who can use the data and the particular purpose of use. Privacy is achieved by the Active Enforcement component that is similar in principle to the Oracle's Virtual Private Database concept. The Active Enforcement component rewrites the incoming query to reflect the privacy restrictions so that it could be safely executed on the database (e.g. by hiding some fields).

The HDB system provides also an auditing mechanism to facilitate the identification of privacy leaks. The system logs any query issued by the data users (only the textual formulation) with some metadata (e.g. the query issuer). In addition, an history of all the modifications performed on the database is maintained to allow the reconstruction of the state of the database at any point in time during the HDB lifetime. If anomalous disclosure of protected data is suspected, the database administrator writes a query reflecting the particular data to be audited. The system identifies in the query logs the set of past inquiries that may be responsible of the unwanted information disclosure and re-executes them over the database once it has been brought back to the state valid at the time of their past execution. On the base of the results obtained, the database administrator could identify if there is any responsibility from the systems and its users in the unlawful data disclosure. This solution works when all the information to be protected is centralized in a single database but is not applicable in our scenario in which different systems should be integrated without saving the data in a single centralized database and consequently a more distributed solution should be devised.

The techniques presented so far are in principle applicable to query management and publication of detailed sensitive data from any database model. However, in the field of business intelligence and data warehousing there are specific means to explore the data (e.g. OLAP, On-Line Analytical Processing) that needs to be considered also from a pri-

vacy point of view. In OLAP systems, a key problem is that of protecting information at low granularity (or projections along certain dimensions) to avoid the inference of sensitive information while aggregations at higher level of granularity could be freely accessed. In [153] it is proposed a way to specify fine-grained authorizations in data cubes partitioned both vertically, depending on the aggregation level (e.g. by year, semester, month or day), and horizontally accordingly to some dimensions (e.g. time, kind of drug, or locations).
Privacy preserving data mining techniques [27, 8] try to make more difficult for an attacker to guess sensitive information starting from the mined results. A typical privacy preserving data mining approach, data perturbation, modifies the data in input adding noise in such a way that their statistical distribution and patters are still preserved and mined results are not compromised even if derived from altered data. Also cryptographic techniques could be used to carry out the mining task from multiple parties in input so that they should not expose themselves as, apart from the mined output, nothing else is learned on the inputs [152].

As said in Chris Clifton et al. [54] and in the PRIVATE-IYE (PRIvacy PreserVing DAta InTEgratIon SYstEm) framework [35] privacy in current information integration systems is an issue limited to already integrated data.
By relaxing this assumption some interesting problems arise for the design of a privacy preserving information integration system. As a preliminary step in many data warehousing and information integration systems data transformations are required to consolidate the data or even to combine the results coming from different sources. Basic information integration approaches as schema matching should be conducted preserving the privacy of the sources that may restrict the visibility of their internal schema and data.
Similarly, duplicate resolution carried out to clean data coming from different sources should preserve privacy of the users and of the data sources the duplicates may originate from [50]. Some privacy-preserving data integration frameworks have been proposed to comply with privacy policy not only on the source data but also when data with different origins is integrated [35, 50]. The PRIVATE-IYE framework proposes a federated form of integration that allows to preserve privacy at two levels: at the source, to make sure the results obtained from the source does not expose protected information; at the integration level, by checking the result obtained combining the fragments coming from the different sources does not violate privacy of some of them. They propose an idea of the components and functionalities of the framework but without implementing it.
As our approach to define privacy policies by samples works with table views it is not limited to a single source but it allows to define privacy constraints on views combining joined tables owned by different schemata. The privacy policies derived can be employed

in a data warehouse system.

The formalization of background knowledge is fundamental to tailor a privacy preserving technique and to find a good balance between protection and utility and to measure the level of privacy achieved [42, 114, 65]. The more information an attacker may exploit to discover the sensitive data the more restrictive should be the privacy protection mechanism reducing the utility of the released data. Our approach allows to define only the constraints that are strictly necessary and can be applied only to the data that will be exposed instead of the whole database. Once data are released it is up to the destination, that is a trusted party, to control they are used properly. For that reason our EHR system is not dealing with collusion attacks on the released information.

2.4 Conclusions

The design of an EHR requires a careful choice of the technology and implementation strategy to satisfy the legal restrictions imposed by the privacy regulations still with a solution that is usable for the end-users. A Data Warehouse solution is not feasible as: it requires to duplicate sensitive data outside the data controller and this is not allowed by the privacy regulations for EHR; data flows only from the sources to the DWH while the reverse is difficult and makes it hard to get a full interoperability among the systems; the data synchronization occurs periodically while in the healthcare domain is important to access to the most recent data in a timely way.
An EII solution poses too strong requirements on the data sources that we cannot impose on systems belonging to different administrative domains. Furthermore, such solution is strongly dependent on the performances and availability of the data sources: a wrong or incomplete answer due to unavailability or inconsistency among the sources is not acceptable in healthcare.
Event-driven SOA mitigates these problems providing to the data sources a solution that is flexible, responsive and loosely coupled and that seems to be the more suitable approach to realise an EHR. However, both design and run-time behaviour of an event-driven SOA system needs to comply with the privacy regulations to guarantee to data controllers and data owners full control on the data shared with the subscribers.

Various methods have been developed to address privacy when sensitive information is disclosed to third parties and processed out of the direct control of their providers. Current privacy policy languages allow to express general privacy requirements without going into the details of the techniques used to process or distribute the data. They give to data providers a way to express which are the authorized purposes for the use of their data. Purpose-based access control mechanisms allow to enforce such purpose-based access control restrictions on the actual data. Privacy policy languages are of general applicability and can be used in different contexts where data are released to third parties. However, their generality makes them sometimes unsuitable to express actionable privacy requirements that are directly "testable" and "verifiable" along the data lifecycle. Errors in capturing the intentions of the data sources and data providers with the definition and implementation of the privacy requirements are discovered only when the system is released and is too late to avoid the disclosure of sensitive data.

Techniques used to preserve privacy in databases and information management systems, as sanitization and anonymization, require a certain level of expertise to the user to identify the set of fields on which they should be applied and to properly balance information loss and level of privacy achieved. Hence they are not affordable for the average user. Moreover, they should act in concert to access control techniques to limit the visibility of certain aggregated data to the privileges and needs of the information consumer.

Scalability in the number of sources to be managed is another critical aspect that may arise in using such techniques so that they are not suitable in outsourced and rapidly evolving environments where privacy requirements may change and the system should be adapted with reduced downtime.

Approaches that do not necessarily require the sanitization of the data can support only limited classes of queries, as in [109], or they are not thought for cross-organisational privacy requirements. The Hippocratic Database [10] is an interesting solution that can work when fed by a single data source (e.g., a hospital) with its set of privacy restrictions and preferences. However, when multiple sources are involved like in an EHR each maintaining the ownership on the data and a specific set of privacy requirements, it is difficult or even not possible to combine, transform and load all their preferences and their data into a central repository. In addition, the HDB automatic query rewriting approach to enforce privacy restrictions, carried out transparently from the information consumers, may lead to inconsistent results in reports when designed by non highly skilled IT users. Information consumers will get misleading results because obtained from the limited view on the data they are authorized to see from the system without any indication on what is happening at query rewriting time [139]. This is particularly critical when data in the EHR are consumed by the medical staff to perform healthcare activities or by the governing bodies to take decisions on the financial plan. For instance if the municipality is not aware only the patients that have given their consent to the EHR are considered by the queries, the results obtained will be an underestimation of the real health state of the town.

Auditing is another fundamental aspect highlighted in HDB [10] we should consider also in developing an EHR. HDB allows the analysis of the query history to identify disclosure of protected information but requires the expertise of the database administrator to carry out the auditing activity. Auditors from the public administrations or institutions certifying the correct implementation of the the privacy regulations in the EHR needs an auditing solution usable even with poor knowledge on the internal database schema of the EHR and capable of exposing an abstract view of the complex parts of the system. It should be possible to explain to the auditors what the system is doing to be compliant to the privacy regulations using abstract models of the underlying enforcing system with its data model, cleaning processes and reporting activities managing in that way the whole compliance lifecycle. In this regard, the work on modeling regulations and on augmenting goals model to justifying requirements with argumentations [141, 95, 22] can be useful to prove to an external auditor (like the privacy guarantor office) that the design of the system is compliant to the privacy regulations.

Chapter 3

Privacy-Preserving Electronic Health Record

This chapter presents an *event-driven SOA* solution for the design and development of an Electronic Health Record. It shows how we tailored the solution to the specificities of the social and medical domains and particularly to the privacy requirements impacting on the design and run-time behavior of the system (Section 3.1). It describes how restrictions derived from privacy regulations are addressed by means of an incremental and fine-grained control of the data distributed among the parties (Section 3.2) according to privacy policies defined by data providers (Section 3.3) and enforced at run-time (Section 3.4). A concrete application of this solution is presented in Chapter 5.

3.1 EHR Architecture

In designing an EHR system there are some legal, technological and organizational constraints of the domain that influenced the decisions on the design and development in order to: minimize the commitment of the partners to join the infrastructure and facilitate the exchange of information minimizing at the same time the traffic; satisfy the privacy regulations in the health domain preventing the duplication of sensitive data outside the boundaries of the data controllers. In this section we explain how we designed an interoperability solution satisfying such constraints.

According to the privacy regulations [57] the actors in this scenario can be classified into:

- **Data Subjects**: citizens and patients, that are the subjects of the personal data;
- **Data Controllers**: socio-health service provider, that collects people's data, including sensitive data, determines the purposes and processing methods of personal data including security matters;

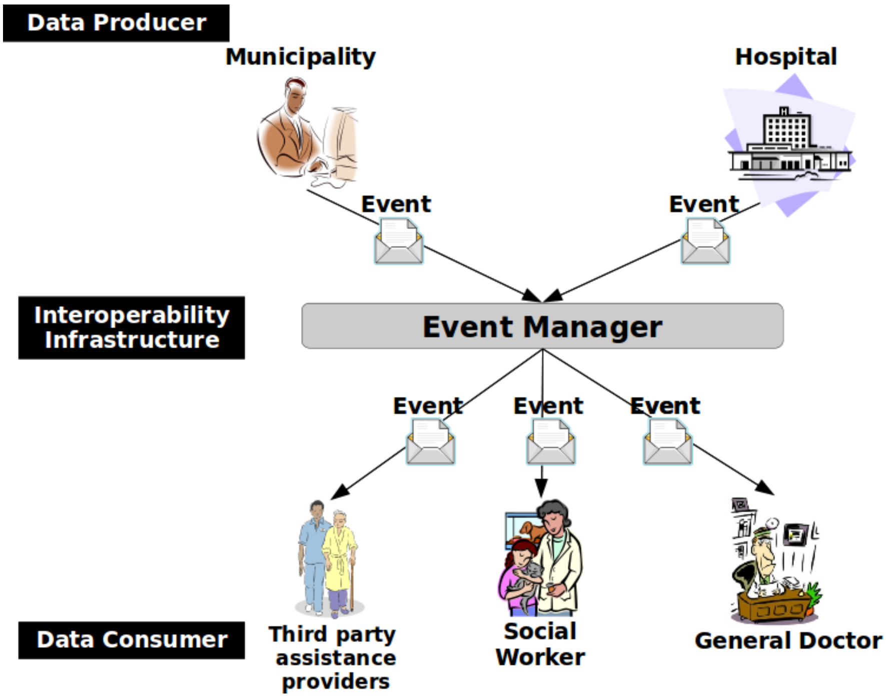


Figure 3.1: General event-based architecture for the cooperation of healthcare and socio-assistive systems.

- **Data Consumers**: all the project partners (for most provincial and governing bodies) having a contract with the data controllers to access and use their data;
- **Data Processor**: the Electronic Health Record processing the data on the controller's behalf.

As shown in Figure 3.1 in our EHR architecture information is exchanged among the parties in form of events. Intuitively, an *event* is the occurrence of a change in the state of a data source that is of interest to other parties. It contains contextual information (like the author of the change, on who it is performed, for what reason, i.e. the type of event, and when) with a payload representing what happened (e.g. the outcome of a clinical examination). In our analysis we modeled the processes going on among the actors in Figure 3.1 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 as we do not dig into the details of the internal databases at the data sources.

An event-driven SOA solution can deal with the technical and organizational restrictions imposed by the scenario but it introduces also some points of attention. The possibility to reach all the subscribers with events is a very powerful communication means, see Figure 3.1, but it can also disseminate easily sensitive information without any control.

Instead, medical and social data, due to their sensitive nature, requires to control not only which events are shared as in a classical publish/subscribe system but also which data inside the events the integration system is authorized to communicate.
The privacy regulations imposes also some restrictions on the way sensitive data can be persisted in the integration system. The Guidelines on the Electronic Health Record and the Health File [88] by the Italian Privacy Guarantor Office impose some restrictions on how personal data and clinical events can be treated and also on how the EHR should be designed. Below are reported some excerpts from these guidelines and their impact on the design of the EHR:

> *"The Electronic Health Record should be set up by* ***prioritizing solutions that do not entail duplication of the medical information*** *created by the health care professionals/bodies that have treated the given data subject."*

This excludes a DWH solution or any alternative solution with a central repository to store the data from the sources.

> *"Secondly, since the medical data and documents contained in a EHR are collected from different sources, the appropriate measures should be taken to* ***allow tracing back the entities responsible for creating and collecting the data and making them available via the EHR - also with a view to accountability****."*

The second statement requires to trace the origin of the data to perform auditing and accountability.

> *Regarding the EHR, since separate clinical records are at issue, it should be ensured that* ***each entity that has created/drafted those records continues to be****, as a rule,* ***the sole data controller in their respect*** *- even though the information is made available to the other entities that are authorised to access the EHR. Availability is often achieved, for instance, by allowing* ***all the entities that have treated the given data subject to share the list of the relevant clinical events; such list is at times set up in the form of an index and/or a list of pointers to the individual clinical events****.*

This requirement states that when multiple institutions are involved in creating and collecting the data that will be used to feed the EHR, it should be guarantee that such institutions maintains the control of their data (i.e. they are the sole data controllers). This statement gives also an hint on the way the EHR can be internally designed with an index of pointers to the individual clinical events.

> *To safeguard data subjects, the purposes in question should accordingly only consist in* ***prevention, diagnosis, care and rehabilitation of the given data subject and exclude any other objective - in particular planning, managing, supervising and assessing health care activities, which can actually be performed in several circumstances without using personal data.*** *This is without prejudice to any requirements arising under criminal law.*
>
> ***If administrative purposes are to be also achieved via EHRs and/or the HF*** *and such purposes are closely related to providing the medical care requested by the given data subject - e.g. as for booking and paying for a given medical examination -* ***the tools in question should be organised in such a way as to keep administrative data separate from medical information. To that end, different authorisation profiles may be allocated as a function of the different operations to be performed.***"

The last two points restrict the purposes for which an EHR can be created and used that is only for prevention, diagnosis, care and rehabilitation. The use of an EHR for administrative purposes is allowed but only if such use is devoted to the delivery of some medical care. In this case the administrative data should be kept separated from the medical data (even physically separated meaning that it should be stored in different databases or tables) and different access profiles should be provided depending on the user and purpose of access. For example, the administrative staff should not have access to the medical data while the medical staff should have visibility of any data in the medical profile of the patient.

The constraints above require to carefully design the events transported to populate the EHR and to apply the event-driven SOA system in such a way that full control on the shared events is guaranteed without the need to store sensitive data.
Etzion and Niblett [67] define an event as: *'an occurrence within a particular system or domain; it is something that has happened, or is contemplated as having happened in that domain'*. As such an event may contain also sensitive data, for example: the admission of a patient to a certain medical division may reveal his diagnosis or may contain the cost of the services used.

In our EHR solution, we explicitly separate the part of data to be considered sensitive from the public data, by defining two kinds of messages carrying the event's content characterized by different levels of sensitiveness and completeness:

- **notification (message)**: is used to signal that an event has occurred in a legacy system. It contains only information on the context in which the event occurred and in particular: the *data subject* (patient/citizen); *what happened* (which type of event and which process of assistance the event is related to); *when* (date of generation in the source system and date of occurrence in reality); *who* generated that information (the data producer organization and its system). It contains the identifying information of a person but not sensitive information (see Figure 3.2).
- **detail (message)**: contains all the data to fully characterize the event that, by default, should be *kept secret* and shared only with explicit authorization of the data producer (e.g. the result of a clinical trial or the report of a psychological analysis) as shown in Figure 3.3a.

Notice that an event is physically transmitted on an event channel as a message containing a serialized form of the event [67] (e.g. in XML). Albeit the two concepts are different, for the sake of simplicity we will use message and event interchangeably. A data event signals a change of state in a source system that is of interest to the other parties (e.g. the completion of a clinical exam by an hospital is of interest to the family doctor of the patient) and should be traced and notified to them. The composition of data events on the same person produced by different sources gives her social and health profile the caregivers need to take care of her.

The distinction between notification and detail is important because it allows to treat the data differently depending on their level of sensitiveness according to the privacy regulations and impacts on the design of the system.

The intuition behind this strategy is the following. It is like if the profile of a person is represented by a sequence of "snapshots" (the events) and each snapshot has a short description of meta-data that explains where, when and by whom the "photo" was taken and what's the picture about (i.e., notification message); the picture is the detail (the detail message) and you can see part of it only if the owner of the picture gives you the permission. A consumer will ask for the detail only if necessary based on the short information in the notification.

This approach allows us to couple the benefit of a pub/sub event-based system (decoupling of publishers and subscribers) with a privacy approach that is compliant with the privacy laws typically adopted in managing healthcare information.

Figure 3.2: Example of notify message.

Field	**Value**	
EventID	129845	Header
Sender	HospitalTN	
SenderURL	http://apss.tn.it	
Name	Anna	Personal data
Surname	Rossi	
Date Of Birth	06-12-1944	
Place Of Birth	Trento	
Residence	Povo	
Fiscal Code	DFGMST68A82H612H	
Date	11-02-2012 22:45	Description
Event type	social evaluation	
Activity	Elderly House Access	
ServiceProvider	Central Elderly House Trento	

Figure 3.4 shows more in details the architecture of the EHR. Sensitive data is maintained at the sources in an *Event Repository* at the *Local Cooperation Gateway* module by the data producers that are the sole data controllers of their data.
The central data processor stores only references (or metadata) to the sensitive data in an *Event Index* that acts as a registry of meta-data. The sensitive information is retrieved on demand by authorized consumers from the sources only when it is necessary. The information is encapsulated as events that are used to move and share information among the legacy systems. The infrastructure driving events from the sources to the destinations is SOA based implemented with web services on top of an enterprise service bus that allows the distribution of events to all interested parties.

The data processor mediates the communication among all the parties and acts as a bridge for the routing and distribution of the events. It contains some domain specific

Field	**Value**
Event ID	129845
Sender	HospitalTN
Sender URL	http://apss.tn.it
Evaluation Outcome	Admitted to RSA
Preference 1	Povo
Preference 2	Trento
Autonomy Level	60%
Cognitive Level	80%
Assistance Network	poor

(a) Detail message at the data producer.

Field	**Value**	
EventID	129845	Header
Sender	HospitalTN	
Sender URL	http://apss.tn.it	
Evaluation Outcome	Admitted to RSA	Details
Preference 1	Povo	
Preference 2	Trento	
Autonomy Level	[redacted]	
Cognitive Level	[redacted]	
Assistance Network	[redacted]	

(b) Filtered detail message.

Figure 3.3: Example of detail message.

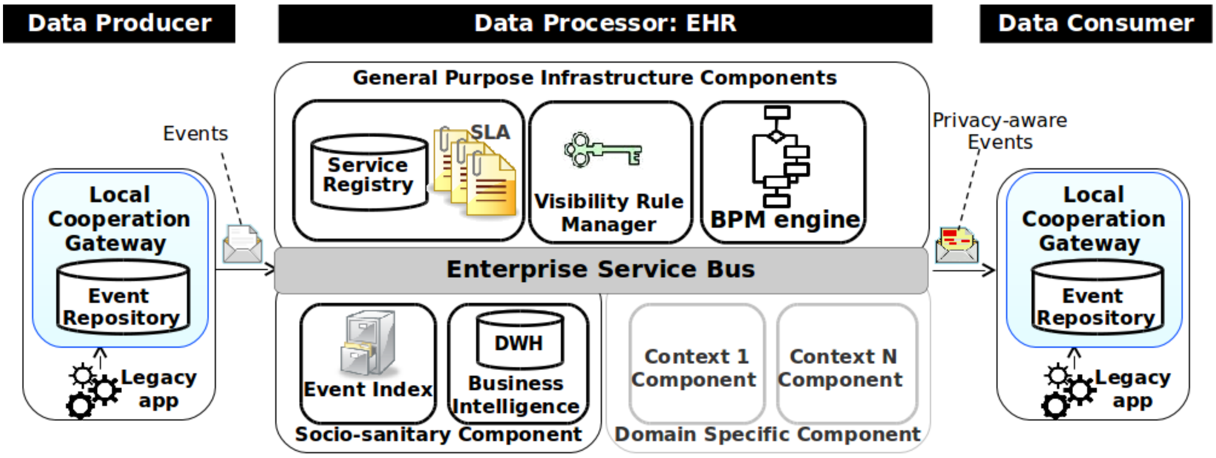


Figure 3.4: Detailed event-based architecture for the cooperation of healthcare and socio-assistive systems.

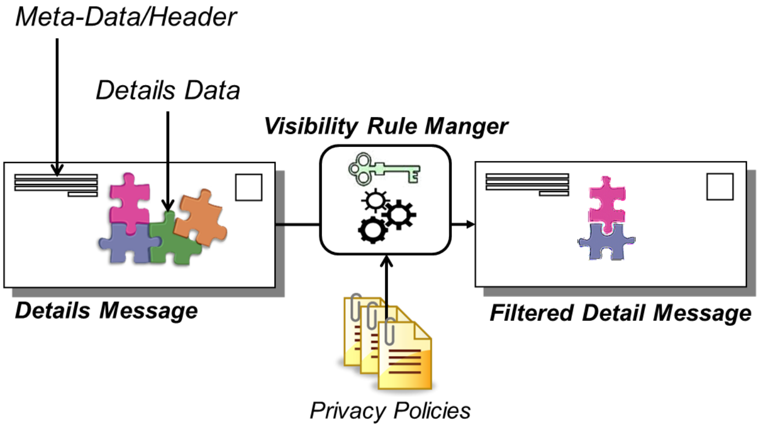


Figure 3.5: Filtering of details message by privacy policies at the Visibility Rule Manger.

components: an index of socio-medical events (the *Event Index* mentioned above) and a business intelligence module to produce reports on the delivered socio-medical services. These components are fed using some general purpose components to manage the list of publishable events (*Service Registry*) and the policies regulating how information is communicated and shared (*Visibility Rule Manger*).

The data processor acts as a broker between data sources and consumers and is the guarantor for the correct application of the privacy policies for retrieving the details and exploring the notifications. The privacy policies allow to restrict the access to the content of the detail messages by removing the fields the requester is not authorized to see as shown in Figure 3.5 and in the resulting filtered event in Figure 3.3b. The values of fields not accessible to the specific consumer for the role and purpose specified are not included in the returned detail message.

In addition, special policies are defined to control the routing of the notifications only to the subscribers authorized by the data owner. The structure and semantics of the privacy policies is presented in details in Section 3.2.
The data processor is the central rooting node of the interoperability infrastructure and it maintains the Event Index, implemented according to the ebXML [37] standard. It stores all the notification messages published by the producers and notifies automatically to the subscribers the events they previously subscribe to.
The detail messages are maintained only in the producer's system as it is the owner and responsible body for that sensitive information. They are also received and processed by a Data Warehouse (DWH) module which is subscribed by default to all event types generated by the sources. In order to be compliant to the security measure imposed by the privacy regulations [88] the identifying information in the notification message like *Name*, *Surname*, *Date of Birth* and *Fiscal Code* are stored in the DWH in encrypted form using an hash function or cryptographic algorithm (e.g. the AES 256 encryption provided by the crypto library in the standard Sun JDK 1.6 distribution [104]) to transform the real data into a unique not invertible code.

The splitting in notification and detail events allows:

- to conceal sensitive information at the data producers with a tight control on its distribution;
- to centralize only the meta-data on the occurrence of events, i.e. notifications, that is not sensitive and can be stored in the Event Index with no violation of the privacy laws and directives [88, 57] which disallow data duplication outside the boundaries of its data controller;
- to tune and differentiate the distribution of notifications and details with just an on/off access control for the notifications and a fine-grained access control for the details;
- to selectively subscribe and access only to the events of interest minimizing the traffic and the effort to join the system as only the events notified and corresponding details should be parsed.

Figure 3.6 and Table 3.1 summarize the interactions taking place among the different system modules to publish and notify to a subscriber a notification event and to retrieve the corresponding details.

Notice how all the interactions (notification with steps 2–3 and details retrieval with steps 4–8) pass through the Interoperability Infrastructure which guarantees the application of the privacy policies to filter the detail message before it is returned to the consumer.

The operations performed internally by the Interoperability Infrastructure are shown in the sequence diagram in Figure 3.7 and further described in Table 3.2.

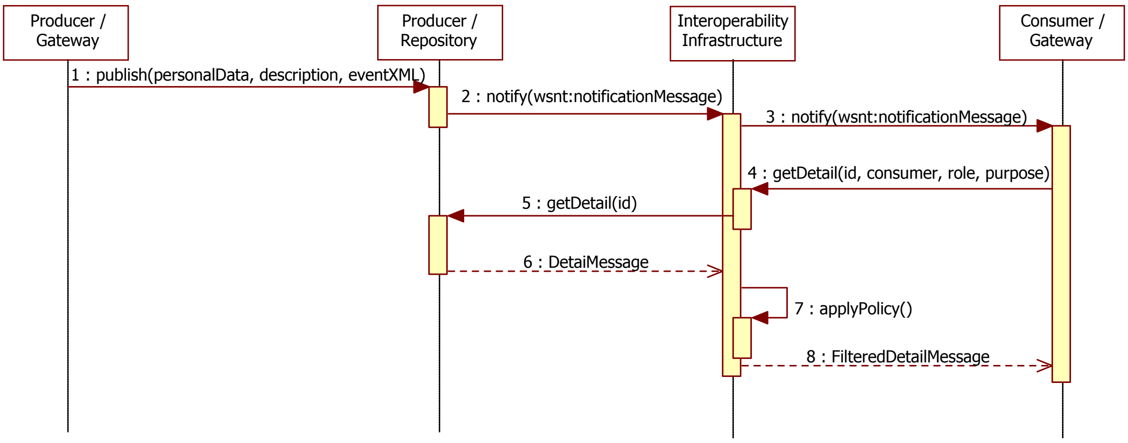

Producer / Gateway
Producer / Repository
Interoperability Infrastructure
Consumer / Gateway
1 : publish(personalData, description, eventXML)
2 : notify(wsnt:notificationMessage)
3 : notify(wsnt:notificationMessage)
4 : getDetail(id, consumer, role, purpose)
5 : getDetail(id)
6 : DetaiMessage
7 : applyPolicy()
8 : FilteredDetailMessage

Figure 3.6: Interactions among the EHR modules for event publication, notification and details retrieval.

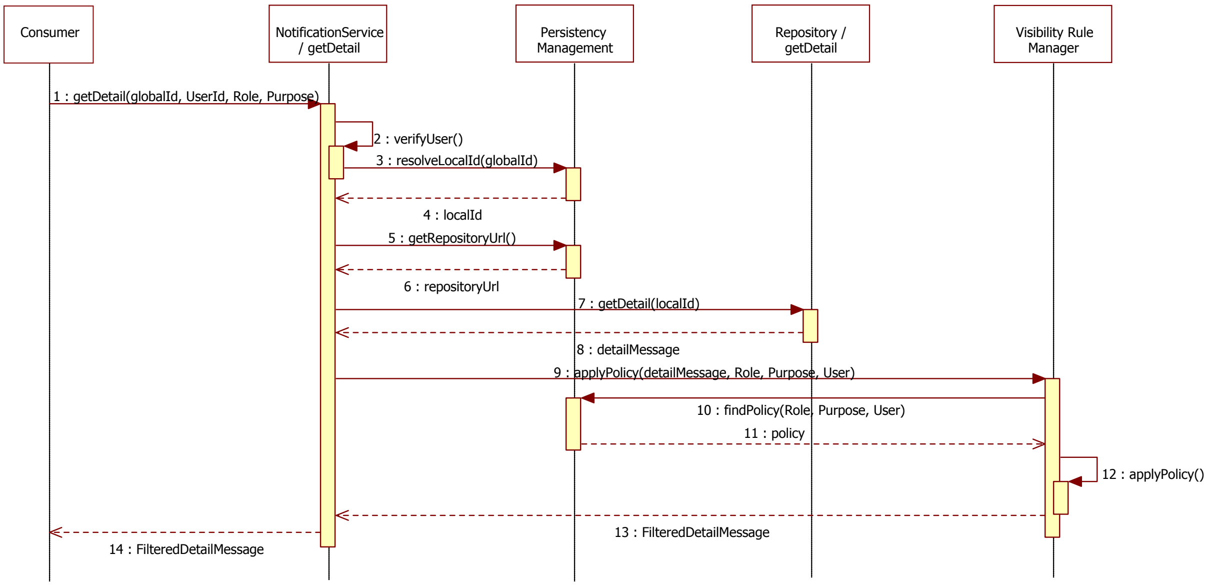

Consumer
NotificationService / getDetail
Persistency Management
Repository / getDetail
Visibility Rule Manager
1 : getDetail(globalId, UserId, Role, Purpose)
2 : verifyUser()
3 : resolveLocalId(globalId)
4 : localId
5 : getRepositoryUrl()
6 : repositoryUrl
7 : getDetail(localId)
8 : detailMessage
9 : applyPolicy(detailMessage, Role, Purpose, User)
10 : findPolicy(Role, Purpose, User)
11 : policy
12 : applyPolicy()
13 : FilteredDetailMessage
14 : FilteredDetailMessage

Figure 3.7: Internal details of the getDetail operation.

Interaction	**Description**
1. publish(personalData, description, eventXML)	The Producer publishes the events (with the personal data of the notification and the details in eventXML) to the event Repository
2. notify(wsnt:notificationMessage)	The Producer notifies the notification message invoking the notify service exposed by the ESB of the Interoperability Infrastructure
3. notify(wsnt:notificationMessage)	The Interoperability Infrastructure distributes the notification messages to the subscribers invoking the notify service exposed by their respective gateways
4. getDetail(id, consumer, role, purpose)	The Consumer asks for the details of a certain notification specifying: event's id (extracted from the notification), the Consumer asking the data, its role and purpose of access
5. getDetail(id)	The Interoperability Infrastructure obtains from the Event Registry the source of the events and forwards it the request for details
6. DetailMessage	The detail message is retrieved and returned
7. applyPolicy()	The Interoperability Infrastructure finds the privacy policies matching the type of event requested, the requester, role and purpose
8. FilteredDetailMessage	The Interoperability Infrastructure applies the privacy policy to the event and returns the filtered event to the requester.

Table 3.1: Interactions for event notification and details retrieval (Figure 3.6).

When a new data controller joins the infrastructure it signs an agreement with the data processor and provides to the service registry the list of events it agrees to share. Another member of the infrastructure willing to get information from the Event Index can explore the catalog of events at the Service Registry and subscribe to the events of interest.
It is up to the data controller owning the events to accept the subscription and to define which portion of details will be accessible to the consumer for specific roles and purposes.

Interaction	**Description**
1: getDetail(globalId, UserId, Role, Purpose)	The Consumer identified by UserId requests the detail of an event with id globalId specifying its Role (e.g. Doctor) and Purpose of use (e.g. medical treatment)
2: verifyUser()	The Notification Service module at the Interoperability Infrastructure verifies the user is subscribed to the event
3: resolveLocalId(globalId)	The Notification Service module asks to the Persistency Management the id of the event in the source system (localId)
4: localId	The Persistency Management returns the local id of the event
5: getRepositoryUrl()	The Notification Service module asks to the Persistency Management also the URL of the source system event repository (repositoryUrl)
6: repositoryUrl	The Persistency Management returns the URL of the source system event repository
7: getDetail(localId)	The Notification Service asks to the repository at the producer the detail message identified by localId
8: detailMessage	The Repository returns the detailMessage with all the data
9: applyPolicy(detailMessage, Role, Purpose, User)	The Notification Service asks to the Visibility Rule Manager to apply the privacy policies valid for that type of event, user performing the request, role and purpose of use of the data
10: findPolicy(Role, Purpose, User)	The Visibility Rule Manager asks to the Persistency Management the policy matching the request
11: policy	The Persistency Management returns the matching policy
12: applyPolicy()	The Visibility Rule Manager applies the policy to the event
13: FilteredDetailMessage	The Visibility Rule Manager returns to the Notification Service the filtered event
14: FilteredDetailMessage	Finally the filtered event is returned to the Consumer.

Table 3.2: Interactions performed by the data processor's modules to retrieve and filter a detail message (Figure 3.7).

The purpose taxonomy for the health domain is well defined at national level by the guarantor office. Before starting to use personal data, a data controller has to notify [58] to the Data Protection Authority (DPA) any processing operation on personal data by means of a notification containing: the categories of data subjects and the type of information related to them that are collected, the purposes of use, the dissemination means and the data recipients. The DPA defines a notification template and a general classification of the types of information to use in the notification that covers well the data types and processing means in our EHR system. However, it does not provide any pre-defined list of user's roles at the data recipient. In our EHR implementation we let to data controllers and data consumers to agree on a list of roles during the subscription phase with a manual interaction and negotiation process among the parties. This solution has some weaknesses that should be studied more in the future as it may lead to a proliferation of roles that could be difficult to control. In this regard, the purposes of use may suffice to deal with basically all the types of data usage performed by consumers and they are more easy to control compared to roles that instead may change dynamically.
Another point of attention for a future extension of the EHR system at national level is represented by the problem of identity management. Advanced federated identity management techniques, like the one under definition in the ICAR project to identify uniquely an individual "*regardless the authentication mechanism employed in the particular domain in which it works*" [49], should be considered.

The data processor receives notifications from each data controller and forwards a copy to the Event Index and to all interested consumers via a publish/subscribe mechanism managed by an *Enterprise Service Bus* (ESB) that can be equipped also with a BPM engine to automate and compose business processes.
Notifications are sent only to authorized consumers that can ask more details for specific purposes. This allows a fine-grained access control and allows the data source to hide part of the details to certain consumers depending on their functional role in the organization (e.g. social welfare department or radiology division) and purposes of use. The data processor is in charge of applying the privacy policies to retrieve only what the consumer is authorized to see from the producer. It also offers the following services and functionalities:

- support both data producers and data consumers in joining the interoperability infrastructure and in particular: the data producer declares the classes of events it will generate in the event catalog and defines the privacy policies for their use by means of the visibility rule manager; the data consumer subscribes to the classes of events it is interested in;

- receive and store the notification messages and deliver them to the subscribers by means of a service bus (a customized version of an open source ESB, *ServiceMix*[1]);
- resolve request for details from the data consumer by enforcing the privacy policies and retrieving from the source the required and accessible information;
- resolve events index inquiry;
- maintains logs of the access request for auditing purposes.

The decoupling between notification messages and detail messages is not only 'structural', as they are carried in different XML messages, but also 'temporal': typically a data consumer gets from the Events Index (either by automatic notification or by querying the index) a notification event and only at a later time it asks for the corresponding details. Requests for details may arrive to the data controller even months after the publication of the notification. Furthermore, medical data requires very long retention period as it should be granted accessibility for the whole lifetime of the patient.
This requires to the data producer the capability to retrieve at any time the details associated to a past notification. These functionalities are encapsulated in the *Local Cooperation gateway* provided as part of the interoperabilty infrastructure to further facilitate the connection with the existing source systems. It is basically a wrapper with a local event repository to persist each detail message that has been notified so that it can be retrieved even when the source systems are un-accessible. In this way:

- it is easy for the data controller to join the infrastructure as the whole communication protocol (WS-Notification, WS-Security standards) is managed by the wrapper;
- there is no need to reconstruct the details afterward as an exact copy of the details is stored in the wrapper at the time of the notification so that the copy can be returned at any time to answer request for details improving also the availability of the system and reducing the impact on the existing source systems.

Usually, publish/subscribe mechanisms are used to deliver information to multiple destinations by decoupling both physically and temporally the producers from the consumers. In many application scenarios it is not required the delivery of the information to be also reliable. The focus is on notifying information at the right time to all the interested parties. A typical example is a stock management system sending stock information to subscribers. The EHR scenario is different as the information delivered is particularly critical and is necessary to collect all the data to have a complete profile of the patients.

[1] ServiceMix, http://servicemix.apache.org/

Even the loss of a single event means to compromise the synchronization with the other parties and consequently the production of an incomplete profile of the patient. For this reason, it is important that the ESB assures a reliable delivery of notifications to the consumers. The ESB used in the project (ServiceMix) does not offer such guarantee. So we extended it with a persistence module: if the destination is unreachable and unable to receive the notifications then the system persists the message and tries repeatedly to send it again until it success. This improves the robustness of the architecture and it makes it usable also in loosely coupled environments in which there is no guarantee of stable connectivity.

All the functionalities mediated by the data controller are logged to support audit activities (internal or from the privacy Guarantor office). A medical record has legal value at least on paper. The same can be expected in the near future for the EHR. For this reason, similar to what happen with the paper based medical record, in the EHR is not possible to delete an event once it has been notified. It is possible to deprecate an event to signal an error in the source data but this will produce a logical deletion that makes the deprecated event un-accessible by the consumers and notifies them of the error. However, to avoid any possibility of falsification, the event will not be removed neither from the Event Index nor the Event Repository but it will remain available for auditing operations.
The publish/subscribe mechanism is implemented with OASIS WS-BrokeredNotification[2] and the Web Services Interoperability (WS-I[3]) standard[4].

3.2 Incremental Privacy on Events

The participation of an entity to the architecture (as data producer or data consumer) is conditioned to the definition of precise contractual agreements with the data processor. The contract between a data source and the data processor constraints how the data could be accessed by a third party and in particular it defines:

- *routing policies*: define which data consumers could receive notifications;
- *privacy policies*: define how many details the data consumer could obtain from a request for details.

These types of policies derives from a study of concrete application scenarios in the healthcare domain and on the privacy regulations imposed by law [88, 57].

[2] http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.htm

[3] http://www.oasis-ws-i.org/

[4] http://www.ws-i.org/profiles/BasicProfile-1.0-2004-04-16.html

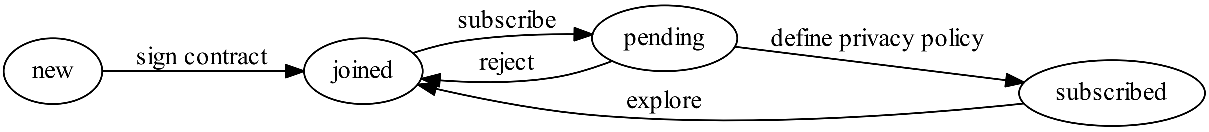


Figure 3.8: Subscription process of a new partner.

The data processor is not able to define such rules as it does not know which part of the event detail is really sensitive and which instead are its safe usages. On the other hand for the data source the definition of privacy rules that can be directly enforced in the system (e.g. in XACML [159]) is a complex and tedious task as it has to do it for each class of event details and requires technical expertise the typical privacy expert does not have.
To facilitate the data sources in this task we support the whole lifecycle of an event from the definition of the privacy policies (both routing and privacy policies) to their enforcement in resolving details requests.
In particular, we provide: a GUI for the intuitive definition of the privacy policies on each class of events (*Privacy Requirements Elicitation Tool*) that produces policies that are directly enforceable in the system; a module that matches a detail request with the corresponding privacy policy (*Policy Enforcer*); and a module to be installed at the sources for the enforcement of the privacy policies on the detail events when a request is authorized (*Local Cooperation Gateway*).

The data producer declares the ability to generate a certain type of event (the *Event Details*) and provides the structure of the event by means of an XSD that is 'installed' in the *Service Registry* module acting as an *event catalog* (see architecture in Figure 3.4). The event catalog, as the structure of its events, is visible to any candidate data consumer that has previously signed a contract with the data processor to join to the cooperation architecture (see Figure 3.8). In order to subscribe to a class of event (e.g. a *blood test*) or to access to its data content, the data consumer (e.g. a *family doctor*) should have the authorization by the data producer. If there is no already a privacy policy defined for that particular data consumer the data producer (that in that case could be the hospital) is notified of the pending access request and it is guided by the Privacy Requirements Elicitation Tool to define a privacy policy. Such privacy policy defines if the *family doctor* has access to the event *blood test* and for which purpose (e.g. for healthcare treatment provisioning) and which part of the event he/she can access (e.g. the results regarding an AIDS test should be obfuscated). For example, an ambulatory can see all the fields of detail messages resulting from a clinical examination but the invoice management system can access only to the billing data; a family doctor has access to all the fields of the event blood test for healthcare treatment provisioning but not to the field with the AIDS test results.

Our approach is innovative for two aspects: i) it allows the data controllers to define their own privacy policies and ii) it gives an incremental control on the access and distribution of sensitive information.

We proposed two alternative implementations for the enforcement of the privacy policies: *centralized enforcement* and *decentralized enforcement*.

In the *centralized enforcement* privacy policies are applied by the data processor on the events retrieved by the data controller. This approach relieves the data controller from dealing with the enforcement of the privacy policies but it is not completely privacy-safe. In this configuration, the events going from the data controller to the data processor contain all the details (even the more sensitive) that are not said to be visible to the requestors. Even if the data processor applies the privacy policies without persisting the data more than the time necessary to apply the policies, there is the risk that the unfiltered event is intercepted and this will increase the probability of privacy violations. In addition, this approach gives to the data processor the responsibility to grant the correct application of the privacy policies and makes it liable in case of privacy leaks. In some cases, the data processor cannot take such a responsibility and the risk of privacy violations could make the approach not completely privacy safe from the eyes of the privacy Guarantor office.

In the *decentralize enforcement*, the privacy policies are applied by the data controller before the event leaves its local Event Repository. This assure that only the data the requester is authorized to see will be delivered to it. This approach gives more guarantees from the privacy point of view even if it requires more work to the data controller to apply the policies. However, the application of the privacy policies could be easily encapsulated in the wrapper module. In practice, it works as follow:

- *policy matching phase*: when a request for blood test details arrives, the data processor finds the privacy policy matching the request and asks to the producer only the fields allowed by the policy. A policy matches a certain request if it refers to the same type of event details and data consumer and if the requested purpose of usage is allowed.

- *policy enforcement phase*: the data controller generates the filtered event for the data processor which delivers it to the consumer.

Only the data accessible to the data consumer leaves the producer (fields not authorized are left empty). In that way, no sensitive information is disclosed neither to the data consumer nor the data controller. The data processor acts as a trusted party maintaining centrally all the privacy policies defined and making sure they are enforced correctly. In both the enforcement approaches if no matching policy is found the request is rejected according to a *deny-by-default* strategy.

The privacy policies defined by the data controllers are stored in the visibility rule manger at the data processor. This assures that there is a single, official place in which privacy policies are maintained, simplifying the synchronization between the parties and, in the future, the identification of conflicts among the policies or particular constraints on them (like the separation of duties). In addition, once the data sources have defined the privacy policies they do not need to keep track of the data consumers and data usage purposes as this is done by the data processor in charge of them.

In the prototype implementation we opt for the centralized enforcement configuration because it was the less impacting on the sources and easier to maintain. In fact, by keeping the policy enforcement detached from the retrieval of detail messages performed by the wrappers at the data producers, we assure that changes and evolutions in the privacy policy definition and enforcement approach do not impact on the data producers.

Notice that we assume the partners are trusted parties and so we do not deal in this work with identity management. In particular, we assume the data processor is under the control of an institution or public body that is officially recognised as trustworthy like for example the Province. However, if the scenario is extended to consider an EHR at national level the assumption of one single official institutions hosting the EHR is no longer valid and is likely to have the outsourcing of the data processing to untrusted parties. In this case, the cooperation infrastructure should be extended with identity management mechanisms that are currently under development at national level with the ICAR (Interoperability and Application Interoperability between Regions) project [49] to: identify the specific users accessing the information, validate their credentials and roles and manage changes and revocation of authorizations in a policy. ICAR proposed the adoption of the SPcoop specifications and the creation of standardized points of contact among the regions encapsulating the security management mechanisms, message encryption and identity management (see PdD in [49]). Our solution is going in this direction but the extension of the EHR at a national level is left to future work.

3.3 Privacy Policy Elicitation

As explained in the previous section, the data consumer defines a privacy policy for each type of event and request of subscription based on the structure of the event detail message. We use XACML to model the privacy policies inside a specific module of the Visibility Rule Manager: the Policy Enforcer module. According to the XACML notation [118], a *policy* is a set of *rules* with *obligations* where a rule specifies which *actions* a certain

subject can perform on a specific *resource*. The *obligations* specifies which operations of the triggered policy should be executed at enforcing time (e.g. to obfuscate part of the resource). In our architecture, an action corresponds to a *purpose* of use (e.g. healthcare treatment, statistical analysis, administration).
A subject is an *actor* reflecting the particular hierarchical structure of the organization. For example, an actor could be a top level organization (e.g. '*Hospital S. Maria*') or a specific department inside it (e.g. '*Laboratory*', '*Dermatology*').
The *role* further specifies the responsibility of the actor in the reference organization (e.g. role of family doctor or social worker or secretary). This multilevel classification allows to define rules giving different access profiles depending on the role of the requestor in the organization. For example, a family doctor can access to all the details about his patients. Instead, a nurse can access only to a limited subset of the detailed information about the same patients.
We consider an *event details* as a list of fields $e = \{f_1, \dots f_k\}$.

Definition 3.3.1. *Let $E(S) = \{e_1, \dots, e_n\}$ and $D(S) = \{d_1, \dots, d_n\}$ be respectively the set of event details and event notifications generated by the data producer S such that e_i has type $\tau_i \in \Gamma(E(S))$ with attributes $\mathbb{A}(\tau_i) = \{a_1, \dots, a_m\}$ for $i = 1, \dots, n$.* ∎

We define *Events Catalog* the set of all the types of event details that the data producers could generate, $E = \bigcup_{i=1}^{n} \Gamma(E(S_i))$.
For each type of event details and type of usage the data producer S defines a privacy policy.

Definition 3.3.2. *Let $E(S) = \{e_1, \dots, e_n\}$ be the set of events a data source S could produce. We define $P_S = \{p_1, \dots, p_n\}$ the set of privacy policies defined by S where $p_i = \{A, O, \tau_i, S, F\}$ such that:*

- *A is an actor that can ask for an event details*
- *O is the role associated to an actor*
- *$\tau_i \in \Gamma(E(S))$ is a type of event details*
- *S is a set of purposes*
- *F is a set of fields where $F \subseteq \mathbb{A}(\tau_i)$.* ∎

Intuitively, a privacy policy indicates which fields F of an event details of type τ_i could be accessed by actor A with role O, for the purposes S. For example, the privacy policy $p = \{$ *National Governance, statistical department, autonomy test, reporting,* $\langle age, sex, autonomy\_score\rangle\}$ allows the *statistical department* of the *National Governance* to access to *age*, *sex* and *autonomy_score* for the event details of type *autonomy test* to perform reporting on the needs of elderly people.

We apply the *deny-by-default* approach so that, unless permitted by some privacy policy, an event details cannot be accessed by any subject. With this rule semantics in mind we used obligations to specify which part of the event details is accessible by a certain subject for some purposes. Notice also that a subject can issue only read requests for an event type.

3.4 Privacy Policy Enforcement

Definition 3.4.1. *Given a privacy policy* $p = \{A, O, \tau_p, S, F\}$ *and an event request* $r = \{A_r, O_r, \tau_e, S_r\}$ *we say that* p *is a* matching policy *for* r *if* $\tau_p = \tau_e \wedge A_r = A \wedge O_r = O \wedge S_r \in S$. ∎

Intuitively a policy matches a certain request if it refers to the same type of event details, actor and role, and if the requested purpose of usage is allowed by the policy.

Definition 3.4.2. *Given the privacy policy* $p = \{A, O, \tau_p, S, F\}$ *and the event instance* e *of type* τ_e *we say that* e *is* privacy safe *for* p *wrt to the request* $r = \{A_r, O_r, \tau_e, S_r\}$*, i.e.* $e \models_r p$*, if* p *is a matching policy for* r *and* $\nexists f \in \tau_e$ *such that* $(e[f]$ *is not empty* $\wedge f \notin F)$ *where* $e[f]$ *is the value of* f *in* e. ∎

Intuitively an event satisfies a privacy policy if it does not expose any field that is not allowed by the policy.
If an event instance e is privacy safe wrt to a request r for all the policies in a set P we write $e \models_r P$ meaning that $e \models_r p_i$, $\forall p_i \in P$.

Privacy policies comes into play in two distinct moments of the events life-cycle and in particular at subscription time and at access time (*request for details* and *event index inquiry*).
In order to subscribe to a class of notification events, the data consumer should be authorized by the data producer, that means there should be a privacy policy regulating the access to the corresponding event details for that particular data consumer. If such a privacy policy is not defined then, according with the deny-by-default semantics, the subscription request is rejected. The inquiry of the event index is managed in the same way, in fact, also in this case the data consumer is asking for notification events.
The request for details resolution is more articulated and we will describe it more in depth with a focus on the specific architectural components involved. A request for details requires to specify the type and identifier of the event to be obtained from the source. This

information is contained in the notification message that is a pre-requisite to issue the request for details and grant that only the data consumer notified by a data producer can access the details. The notification is obtained either automatically by means of the pub/sub service offered by the infrastructure or by direct inquiry of the event index.

Figure 3.9 shows the internal components of the Policy Enforcer module in the data processor which are in charge of: receive the request for details from the data consumer; retrieve the matching privacy policy associated to the Event Type and Event ID specified in the request; apply and evaluate the policy against the request and finally return a response with the result of the authorization decision. The result is an event details with values only for the fields authorized by the matching policy. The components which constitute the Policy Enforcer are based on the XACML Specification.

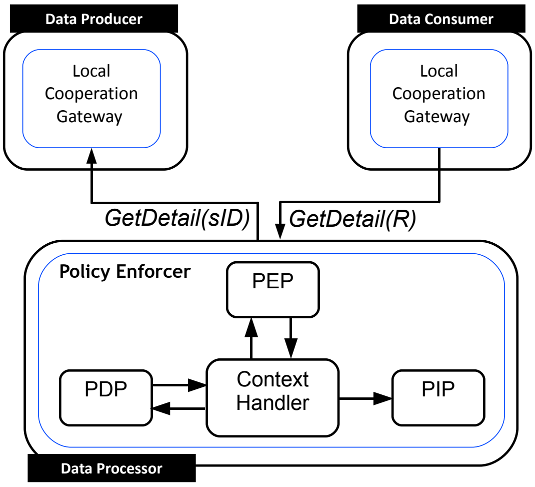


Figure 3.9: Detail request resolution and privacy policy enforcement.

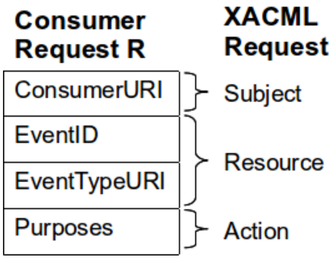


Figure 3.10: Mapping in XACML request notation.

Algorithm 1: $getDetail(R) \mapsto e$

```
Data: $R = \{a, o, \tau_e, eID, s\} \neq \emptyset$
      $P$ set of policies defined by the data producers
Result: $e \models_r P$
$sID \Leftarrow retrieveEventProducerId(eID)$
$\langle A, o, e_j, S, F \rangle \Leftarrow matchingPolicy(R)$
if $(evaluate(\langle A, o, e_j, S, F \rangle, R) \equiv permit)$ then
    $d \Leftarrow getDetail(sID)$
    return $applyObligations(d, F)$
end
return $deny$
```

Algorithm 1 shows the actions performed by the policy enforcer in the *getDetail*(R) method at the data processor to resolve an authorization request R issued by a data consumer a with role o to access to the event with identifier eID and type τ_e for purpose s. The main steps performed by the Policy Enforcer are described below (see Figure 3.9):

1. The authorization request is received by the Policy Enforcement Point (PEP). Through the Policy Information Point (PIP) it retrieves the corresponding local event ID (sID) valid in the data producer of the event. This mapping step is necessary as the event identifier distributed in the notification messages (eID) is a global artificial identifier generated by the data processor to identify the events independently from their data producers.

2. The PEP sends the request to the Policy Decision Point (PDP). The PDP retrieves the matching policy associated to the data producer, the data consumer and the resource: $\langle A, o, e_j, S, F\rangle$.

3. The PDP evaluates the matching policy and sends the result to the PEP. If there is no matching policy for the request or the evaluation fails, the response will be *deny* and an *Access Denied message* is sent to the data consumer.

4. If the matching policy successfully evaluates the request (*permit decision*), the PEP asks the event details (F) to the data producer (i.e. the owner of the resource). The *getDetail*(sID) invocation retrieves the Event Details from the internal events repository at the Local Cooperation Gateway of the producer.

5. Finally, the event's fields that are not authorized are removed by applying the obligations in the policy. The *applyObligations*(d, F) produces the Privacy-Aware Event to be sent to the data consumer.

Notice that only the data accessible to the data consumer leaves the data processor and the fields that are not authorized are left empty. If there is no matching policy, the request results in a deny and no event is returned.

The data processor is a trusted entity which performs the application of policies, traces the request of access and does the message routing between data producers and data consumers.

The architecture of the policy enforcer reflects XACML but the way we interact with the data producer and data consumer is independent from the underlying notation and enforcement strategy. As shown in Figure 3.10 the request for details of the data consumer is mapped to an XACML request by the policy enforcer. As Policy Decision Point (PDP) to evaluate the XACML policies we used the *XACML Enterprise*[5] implementation released

[5] XACML Enterprise, http://code.google.com/p/enterprise-java-xacml/

under Apache License 2.0. It supports XACML v2.0 and provides good performances using efficient policy evaluation mechanisms as show by the comparative analysis in [151]. 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). Furthermore, there is no limit in the number and depth of the privacy policies defined, so that the data producer can define many alternative privacy profiles depending on the data consumer to provide different views on the data (e.g. a social worker or a doctor may want to hide their complaints on the quality of work of a nurse to her but to notify the evaluation to her supervisor).

3.5 Conclusions

This chapter presents an EHR architecture based on SOA and EDA in which data is shared among information systems by means of events. An interoperability infrastructure is devised to allow the cooperation of different entities acting as data producers and data consumers in a completely loosely coupled manner. An event manager acts as a broker to manage the publication and subscription to events of the systems joining the EHR.

The EHR has been designed to be compliant to the privacy regulations regarding EHR and Health File. In particular, it avoids to store any sensitive data about patients but it maintains only public information on the individual in an index used to reconstruct and retrieve the data directly from the data producers. In this way, the data producers act as data controllers and they maintain the full control of the information collected from the data subjects even when shared in the EHR system. The access to sensitive information is controlled by means of contracts and privacy policies defined at subscription time by the data producers and enforced at run-time to assure potentially sensitive data are released only to authorized consumers.

The capability of EDA to reach easily many data consumers by means of a publish/subscribe mechanisms are reconciled with the requirements of tight control on sensitive data coming from the data providers by dividing events occurred to patients into: notifications, reflecting non-sensitive information on the event occurred to the patient; details, containing the complete description of the event occurred to the patient including also sensitive data (e.g. the diagnosis). Notifications are delivered by the broker to all the subscribers to notify them on the evolution of the state of the patient but details are accessible only by explicit request to the data producers (mediated by the interoperability infrastructure) to get only the part allowed by a privacy policy. This approach combines the capability of an event-based system with an incremental, tight control of the shared information by means of fine-grained privacy policies.

Chapter 4

Privacy Compliance Checking

The user of modern computerized systems is constantly shifting towards those that are less technically skilled but at the same time experts in non-technical areas. Those users are naturally facing difficulties in coping with the huge amount of data that the modern repositories typically contain, and require ways to perform their data management tasks within the limited amount of time they have available by using as less effort as possible. A technique that has received considerable attention is that of sampling, that abstracts a large data set into some readable portion.

Although sampling has been used in many application scenarios, mainly to describe the data, in this work we use it in order to propagate privacy constraints into the underlying base tables. In particular, given a dataset, we investigate ways to create a representative data set of specific size that maximizes the number of privacy constraints that can be defined on the original dataset through the sample. In our context, a privacy constraint is a constraints that disallows a combination of data to be visible.

4.1 Motivation

Consider the Electronic Health Record database in Table 4.1. It integrates the medical data about patients in different hospitals and departments at a very detailed level to allow statistical analysis and reporting for the medical staff or the governance.

A report is a query whose outcome is a view over the database (e.g. "select all the female/male patients" or "select patients in a certain age class"). A view may contain sensitive data the patients or the privacy regulations do not allow to expose. Such data represents forbidden values that should be identified and removed (e.g. by masking them) from the view before the report is accessed. The values to be hidden depends on the regulations to comply with, the consent expressed by the data subject (i.e. the patient), the purpose of use of the report, who access the report and its specific data content [125].

Table 4.1: Privacy violation indications.

Tuple	Name	AgeClass	DOB	POB	Sex	Department	Symptom	Diagnosis
1	Sophy	18 – 30	02/10/1983	Meano	F	Physiotherapy	Paralysis	Sclerosis
2	Jeremy	30 – 65	01/11/1966	Povo	M	Physiotherapy	Numbness	Sclerosis
3	Helene	< 18	28/01/1995	Trento	F	Psychology	Raping	Depression
4	Ketty	< 18	13/12/2009	Povo	F	Infectiology	Fever	Measles
5	Rose	30 – 65	06/12/1977	Daone	F	Psychology	Raping	Depression
6	Bob	18 – 30	09/05/1991	Daone	M	Oncology	Anemia	Leukemia
7	Paul	> 65	10/06/1941	Brenta	M	Infectiology	Fever	Mumps
8	Chris	30 – 65	12/5/1975	Lavis	F	STD	Infection	Candidiasis
9	Lidia	18 – 30	02/08/1982	Vela	F	STD	Infection	AIDS
10	Jeremy	30 – 65	07/03/1976	Vigo	M	STD	Fever	AIDS
11	Tim	30 – 65	07/01/1978	Zava	M	STD	Flu	AIDS
12	Marta	18 – 30	05/04/1984	Povo	F	Oncology	Weakness	Cancer
13	Julien	> 65	11/05/1931	Obra	F	Oncology	Anemia	Cancer

Usually the privacy regulations, both in Europe [135, 57, 88] and US (e.g. HIPPA [7, 32], COPPA [71], GLB [70], Data Protection Act [136] and OECD guidelines [131]) requires the data controller entity to designate a privacy expert with the responsibility to implement the regulations. The privacy expert is a person with adequate knowledge of the domain and regulations that given the result of a query can indicate if it contains privacy violations and where.

Tuple	Name	AgeClass	DOB	POB	Department
1	Sophy	18 – 30	02/10/1983	Meano	Physiotherapy
3	Helene	< 18	28/01/1995	Trento	Psychology
4	Ketty	< 18	13/12/2009	Povo	Infectiology
5	Rose	30 – 65	06/12/1977	Daone	Psychology
8	Chris	30 – 65	12/5/1975	Lavis	STD
9	Lidia	18 – 30	02/08/1982	Vela	STD
12	Marta	18 – 30	05/04/1984	Povo	Oncology
13	Julien	> 65	11/05/1931	Obra	Oncology

(a) View on female patients.

Tuple	Name	AgeClass	DOB	POB	Department
2	Jeremy	30 – 65	01/11/1966	Povo	Physiotherapy
6	Bob	18 – 30	09/05/1991	Daone	Oncology
7	Paul	> 65	10/06/1941	Brenta	Infectiology
10	Jeremy	30 – 65	07/03/1976	Vigo	STD
11	Tim	30 – 65	07/01/1978	Zava	STD

(b) View on male patients.

Table 4.2: Example of Sample Table.

As there are as many views as the queries definable on the database, and their number is exponential with respect to database tables dimensions, it is not feasible to ask to the privacy expert to check each view and notify all the violations.

Instead, it is better to perform the check just once on the whole database Table 4.1, and then to propagate the indications of privacy violations to the views [44].

We said privacy violations, and not yet privacy constraints because we cannot expect the user to be able to define the privacy constraints directly on the database due to its complexity when compared to a report. What the user can do is to analyse the tuples one at a time and to indicate the group of attributes that, if shown together, will lead to a privacy violation as shown in Table 4.1: greyed cells denotes privacy violating attributes. But in this way the user will just indicate the group of attributes that are privacy violating but not why they should be considered violating. It may be possible that the user wants to remove just the specific values in the single tuple, e.g., remove attribute $\langle Name \rangle$, when the values in tuple t_8 occurs $(Name = Ketty \wedge AgeClass < 18 \wedge DOB = 13/12/2009 \wedge POB = Povo \wedge Sex = F \wedge Department = Infectiology \wedge Symptom = Fever \wedge Diagnosis = Measles)$. This approach is correct, as it reflects exactly what the user said on the data, but it does not give a description of the privacy constraint in such a manner that it precisely identify the violations, and that it is still generic enough to be applicable also to other views in the same database. For example, the indication of the user in t_4 could be expressed with a shorter constraint like: remove $\langle Name, AgeClass, DOB, POB \rangle$ when $Sex = F \wedge AgeClass < 18$. This new constraint covers the violation indicated in tuple t_4 and also in tuple t_3. So with only one constraint we can express 2 indications given by the user. Generalization is good as far as the resulting constraint is not too much generic, meaning that it selects also tuples that the user would not hide. For example, t_3 and t_4 can be covered just with one constraint with a single condition like: remove $\langle Name \rangle$ when $Sex = F$. But this constraint is not correct as it identifies also t_{12} and t_{13} that are not "forbidden".

We will show some techniques to find a minimal set of constraints capable to capture and to describe user's indications in a way that can be easily propagated to the view derived from the DB.

The interaction with the user is the strong point of this approach, as in this way one is granted to capture user requirements. However, it is also its weakness if the table is very big (that is really common in databases for EHR). The straightforward solution would be to present the user with the whole relation R and let him/her select the privacy violating values. However, this may be infeasible since the relation R may be really large causing many presentation issues. A practical solution is to present the user with portions of the relation R with the hope that they will contain some privacy violating values so that the user could identify them.

These portions of the relation R are nothing more than views. The size of these views is restricted by the capabilities and comfort of the user. Furthermore, the views may be overlapping, which means that a relation value (or a group of values) may be repeatedly presented to the user. Given the fact that the time availability of the user may be limited, it is critically important to present the user with the more representative parts of the relation R that are more likely to contain privacy violating values.

By showing larger views, larger portions of the relation data are covered, thus, less views are needed to show all the data of the relation R. However, the size of modern computer monitors and certain user preferences may pose a further restriction on the size of the views that can be displayed.

The U.S. Government study on the usability of web sites [83] provides some interesting guidelines for the construction of highly usable web sites that can be useful in our problem. A fundamental aspect to take into account in structuring a web page is to make sure all the information can fit in the screen size to avoid horizontal scrolling, requiring the usage of pagination to break up information into shorter pages. Appropriate line length and page dimensions should be chosen to allow the user to find easily the most important information in the web page at a first look and in a short time. Items shown in the pages should also respect an order which allows to find easily the most important information. Given the above, our problem can be formulated as follows: find a set of views from relation R with the aim to capture quickly the constraints the user has in mind.

The privacy violations that the privacy expert signals on the samples need to be turned into privacy constraints. There are three main challenges in doing so. First, the expert scans the sample one tuple at a time, thus any specification derived from her signaling a violation should be considered only for the specific tuple. For instance the fact that the expert signaled a violation of the display of the SSN for a tuple should be interpreted as a constraint for the person represented by the specific tuple and not for everybody: i.e., there is no evidence indicating that this should apply to everybody.

The second challenge is that the expert is typically not providing any justification for the privacy violation, thus the only condition that can be assumed is the *conjunction of all the attributes of the tuple on which the violation is signaled.* For instance, if for the tuple $t = [SSN : 1, name : John, city : LA, genre : M]$ the privacy expert says that SSN should not be shown, then the only constraint valid in any table with the same attributes of tuple t is $(SSN = 1) \wedge (name = John) \wedge (city = LA) \wedge (genre = M)$. This constraint is correct but it does not necessarily capture the reason of the privacy violation the user had in mind. It may be that the user wanted to hide the SSN in tuple t for all the males in LA city regardless the name. So the problem is to derive a concise representation of the privacy constraints equivalent to the *candidate privacy constraint.* A set of concise constraints is not said to be minimal as it may still contain some redundant constraints since a constraint may be implied by another. For example, if the constraint on tuple t hiding the SSN is satisfied by a view, then any other constraint on that tuple requiring to hide SSN together with another attribute in t is satisfied too (e.g. hide SSN and $genre$ in t). This leads to another problem: how to remove redundant constraints because implied by others.

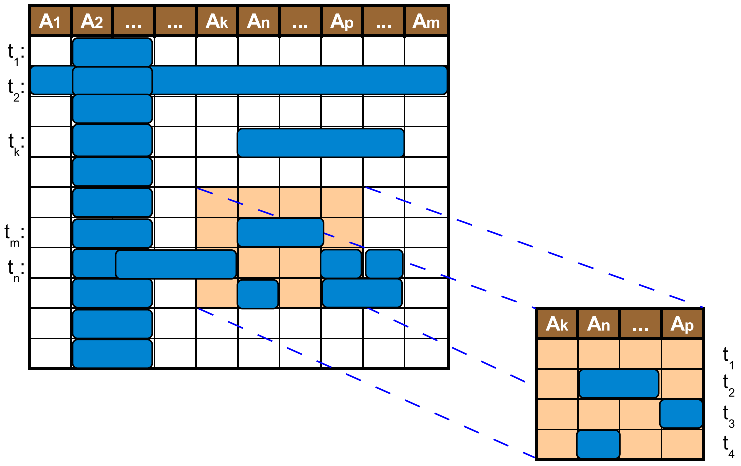


Figure 4.1: Constraint violations indicated on a relation R and an illustration of those constraints that are also applying on a sample of it.

The third challenge is that the expert has typically few time to analyse the data so it is important to show first the data that are more likely to contain the privacy violations. In this way even if the user decides to interrupt the analysis of the data without looking at all the tuples in the view we are confident we were able to capture a good number of constraints.

In this chapter we analyse the three challenges above providing a solution to elicit and derive a concise definition of privacy requirements interacting with the privacy experts by means of views.

4.2 Preliminaries

Let $\mathcal{L}$ be an infinite set of labels, and $\mathcal{D}$ an infinite domain of values. A *relation* is a finite set of tuples of the form $[A_1{:}v_1, A_2{:}v_2, \ldots, A_n{:}v_n]$, where $v_i{\in}\mathcal{D}$ and $A_i{\in}\mathcal{L}$ for $i{=}1..n$. The labels $A_1, \ldots, A_n$ should have no duplications and are referred to as the *attributes* of the relation, and all together form its schema. We will use the term *value* of a relation R, to refer to an attribute A of one of its tuples t, or the actual value v it contains, and we will denote it by $t[A]$. Furthermore, we will often view a relation as a set of values which permits us to use the notation $v{\in}R$ to denote that there is a tuple $t{\in}R$ for which $v{=}t[A]$ with A being one of the attributes of R. The expression $t[AB..F]$ is nothing more than a shorthand for $[A{:}t[A], B{:}t[B], \ldots, F{:}t[F]]$. A *masked value* of an attribute in a tuple is a special value "$*$" used to avoid revealing the actual value of the attribute.

Tuple	Name	AgeClass	DOB	POB	Sex	Department	Symptom	Diagnosis
1	Sophy	18 − 30	02/10/1983	Meano	F	Physiotherapy	Paralysis	Sclerosis
2	Jeremy	30 − 65	01/11/1966	Povo	M	Physiotherapy	Numbness	Sclerosis
3	Helene	< 18	28/01/1995	Trento	F	Psychology	Raping	Depression
4	Ketty	< 18	13/12/2009	Povo	F	Infectiology	Fever	Measles
5	Rose	30 − 65	06/12/1977	Daone	F	Psychology	Raping	Depression
6	Bob	18 − 30	09/05/1991	Daone	M	Oncology	Anemia	Leukemia
7	Paul	> 65	10/06/1941	Brenta	M	Infectiology	Fever	Mumps
8	Chris	30 − 65	12/5/1975	Lavis	F	STD	Infection	Candidiasis
9	Lidia	18 − 30	02/08/1982	Vela	F	STD	Infection	AIDS
10	Jeremy	30 − 65	07/03/1976	Vigo	M	STD	Fever	AIDS
11	Tim	30 − 65	07/01/1978	Zava	M	STD	Flu	AIDS
12	Marta	18 − 30	05/04/1984	Povo	F	Oncology	Weakness	Cancer
13	Julien	> 65	11/05/1931	Obra	M	Oncology	Anemia	Cancer

(a) Privacy Constraints in EHR table.

Tuple	Name	AgeClass	DOB	POB	Sex	Department
1	Sophy	18 − 30	02/10/1983	Meano	F	Physiotherapy
3	Helene	< 18	28/01/1995	Trento	F	Psychology
4	Ketty	< 18	13/12/2009	Povo	F	Infectiology
6	Bob	18 − 30	09/05/1991	Daone	M	Oncology
7	Paul	> 65	10/06/1941	Brenta	M	Infectiology
8	Chris	30 − 65	12/5/1975	Lavis	F	STD
9	Lidia	18 − 30	02/08/1982	Vela	F	STD
10	Jeremy	30 − 65	07/03/1976	Vigo	M	STD

(b) Privacy Constraints in Sample table.

Table 4.3: Privacy Constraints Conditions.

Name	AgeClass	POB	Department	Symptom	Diagnosis
*	*	*	STD	Infection	Candidiasis
Sophy	18 − 30	Meano	Physiotherapy	Paralysis	Sclerosis
*	< 18	Trento	Psychology	Raping	Depression
*	< 18	Povo	Infectiology	Fever	Measles
*	*	*	STD	Infection	AIDS
Marta	18 − 30	Povo	Oncology	Weakness	Cancer
*	30 − 65	Daone	Psychology	Raping	Depression

(a) Diagnosis by Age Class and POB for Female.

Name	AgeClass	DOB	POB	Department	Diagnosis
Bob	18 − 30	09/05/1991	Daone	Oncology	Leukemia
Paul	> 65	10/06/1941	Brenta	Infectiology	Mumps
*	*	07/03/1976	*	STD	AIDS
*	*	07/01/1978	*	STD	AIDS
Julien	> 65	11/05/1931	Obra	Oncology	Cancer
Jeremy	30 − 65	*	*	Numbness	Sclerosis

(b) Diagnosis by Age Class and POB for Male.

Table 4.4: Views satisfying the privacy constraints indicated by the user.

A *privacy constraint* is a query. In this work we focus only on privacy constraints defined by select-project queries for which the `where` clause is a conjunction of conditions of the form $A{=}v$, with $A{\in}\mathcal{L}$ and $v{\in}\mathcal{D}$. This is a class of simple queries that are commonly used in practice [102] but, most importantly, they accurately describe the class of privacy constraints that are used in practice [59, 55]. For brevity, we will use the expression $\ll\mathcal{A}\colon \mathcal{C}\gg$, where $\mathcal{A}$ is a list of attributes $A_1, A_2, \ldots, A_n$ and $\mathcal{C}$ is a conjunction $A'_1{=}v_1 \wedge A'_2{=}v_2 \wedge \ldots \wedge A'_m{=}v_m$, as a shorthand for the privacy constraint specified by the query

```
select $A_1$, $A_2$, ..., $A_n$      from $R$
where $A'_1{=}v_1$ and $A'_2{=}v_2$ and ... and $A'_m{=}v_m$
```

We will refer to the conditions of the form $A{=}v$ as *primitive conditions*, and we will denote the set of primitive conditions in a conjunction $\mathcal{C}$ as $\|\mathcal{C}\|$.

A privacy constraint is said to *apply* on a relation R if every attribute mentioned in the constraint exists in R. We will denote the set of all privacy constraints that apply on a relation R by $\mathcal{P}^R$.

For example the constraint $\ll Name, AgeClass, POB\colon Department = STD\gg$ applies to Table 4.3a but $\ll Name, AgeClass, POB, Phone\colon Department = STD\gg$ does not as *Phone* is not an attribute of that table.

Intuitively, a privacy constraint describes a set of attributes and certain conditions such that if there is a tuple satisfying the conditions of the constraint, the set of attributes that the constraint provides should not be visible, i.e., they are masked by suppressing the values so that they are not displayed. The conditions that need to be satisfied are specified by the `where` clause of the privacy constraint and the attributes by its `select` clause.

Definition 4.2.1. *A relation R satisfies a privacy constraint p :$\ll\mathcal{A}\colon \mathcal{C}\gg$, denoted as $R{\models}p$, if p applies on R and the query* `select` $\mathcal{A}$ `from` R `where` $\mathcal{C}$ *returns only masked values. If not, the constraint is said to be* violated. ∎

We also define the *coverage* of a privacy constraints as follows.

Definition 4.2.2. *Given a relation R and a set of privacy constraints $P = \{p : p = \ll\mathcal{A}\colon \mathcal{C}\gg\}$ coverage of P in R, $\widehat{R}(P)$, is defined as the tuples selected by the privacy constraints in P that is $\widehat{R}(P) = \bigcup_{\ll\mathcal{A}:\mathcal{C}\gg\in P} R(\mathcal{C})$ where $R(\mathcal{C}) = \{t \in R : t \text{ satisfies } \mathcal{C}\}$.* ∎

Example 4.2.1. *Table 4.3a shows some examples of privacy constraints violations where the cells in gray are the attributes the user does not want to see together in a view and the cells in yellow are the conditions identifying the tuples covered by the constraint. For example the constraint $\ll DOB, POB\colon Diagnosis = Sclerosis\gg$ covers the first two tuples and means that the user does not want to see* DOB *together with* POB *when the* Diagnosis *is* Sclerosis*.*

∎

The privacy constraints considered in our examples derive from the regulations on the treatment of personal data for which special protection is needed for underage people,

people affected by sexually transmitted disease (abbreviated with the acronym STD) or victims of raping [57].

Often the specification of two privacy constraints p and p' are such that, if p' is satisfied in R, p is satisfied in R as well. In this case the first is said to be *subsumed* by the second. The subsumed privacy constraints are redundant in the presence of their subsumees.

Definition 4.2.3. *A constraint $p{=}\ll\mathcal{A}\textbf{:}\ C\gg$ in a relation R is said to be subsumed by a constraint $p'{=}\ll\mathcal{A}'\textbf{:}\ C'\gg$, denoted as $p\dot{\leq}p'$, if $\mathcal{A}'{\subseteq}\mathcal{A}$ and $C{\Rightarrow}C'$ in R.* ∎

Example 4.2.2. *For the relation in Table 4.3a the privacy constraint $\ll DOB, POB\textbf{:}$ $Diagnosis = Sclerosis \wedge Department = Physiotherapy\gg$ is subsumed by the constraint $\ll DOB\textbf{:}\ Diagnosis = Sclerosis\gg$.* ∎

The subsumption property can be used to discover redundant constraints when different privacy policies definitions originate from the indications of the expert (especially when different experts are involved in the task).

Definition 4.2.4. *Given a relation R with a set of attributes $\mathcal{A}$, a* sample *T is a relation created by a set of tuples from R projected on a subset of $\mathcal{A}$. The symbol $\mathcal{S}^R$ denotes the set of all possible samples of R. The* image *of a tuple $t_S{\in}S$ is any tuple $t{\in}R$ such that the projection of t on the attributes of S is the tuple t_S. The set of such tuples for t_S is denoted as* $\mathbf{Img}(t_S)$. ∎

Example 4.2.3. *Table 4.3b shows a sample of 8 tuples from Table 4.3a selected on the attributes $\{Name, AgeClass, DOB, POB, Sex, Department\}$. Notice how all the privacy constraints defined on the corresponding tuples of Table 4.3a are also applicable to the sample but not $\ll DOB, POB\textbf{:}\ Diagnosis = Sclerosis\gg$ because the Diagnosis attribute does not appear in this sample.* ∎

Theorem 4.2.1. *If a relation R satisfies a privacy constraint p, any sample of R on which the p applies, also satisfies p.* ∎

Proof. Assume a privacy constraint $\ll\mathcal{A}\textbf{:}\ \mathcal{C}\gg$ and that all the attributes $\mathcal{A}$ are in S. Let a tuple $t_s{\in}S$ satisfying the constraints $\mathcal{C}$. Every tuple in $\mathbf{Img}(t_S)$ will also satisfy $\mathcal{C}$, and since the constraint is not violated in R, the values of the attributes $\mathcal{A}$ of these tuples will be masked. As a consequence, the values of these attributes in t_S will also be masked, since t_S is a projection of one of the tuples in $\mathbf{Img}(t_S)$. Thus, the constraint is satisfied also in S. ∎

Theorem 4.2.1 implies that if a violation is discovered in a sample then it holds also in R. However, the vice-versa is not true as it is easy to create a sample S satisfying a privacy constraint p that instead is violated in R. If we restrict Theorem 4.2.1 on the tuples projected in a sample then we can derive the following lemma.

Lemma 4.2.1. *If a sample S satisfies a privacy constraint p, then also* $\mathbf{Img}(S) = \bigcup_{t_S \in S} \mathbf{Img}(t_S)$ *satisfies p and vice-versa.* ∎

In a typical privacy constraint elicitation scenario for a relation R, the privacy expert is asked to identify any violation of real world privacy constraints. In practical situation, the relation with which the user is presented is a sample of the full relation that exists in the repository. This is because due to technical and practical limitations it may not be possible to present the whole relation R. The expert examines the tuples of the sample, one at a time and highlights groups of attributes in the same tuple whose display violates some real world privacy rule.[1] However, the expert does not specify under what conditions the violation occurs.

For each such specification, a constraint $\ll\mathcal{A}$**:** $C\gg$ can be created, where the set of attributes $\mathcal{A}$ is the set of attributes that the expert indicated as privacy violating. For the condition C, since the expert is not explicitly stating the reason of the violation, we have no other choice than to consider everything that the expert sees from the tuple. This means that the condition will be a conjunction of attribute-value assignments of the form A=v, one for each attribute-value of the tuple in the sample on which the expert signaled the violation.

In doing that we apply a conservative approach as we create a *candidate privacy constraint* based only on the attributes and values seen by the user when analysing the specific tuple.

Example 4.2.4. *Assume that the user sees the tuple* 8 *about patient* Chris *in Table 4.3a and she highlights the group of attributes* $\{Name, AgeClass, POB\}$*, then the corresponding privacy constraint can be expressed as:* $\ll Name, AgeClass, POB$**:** $Name = Chris, AgeClass = 30-65, DOB = 12/5/1975, POB = Lavis, Sex = F, Department = STD, Symptom = Infection, Candidiasis = Diagnosis\gg$*. Note how this constraint is subsumed by* $\ll Name, AgeClass, POB$**:** $Department = STD\gg$*.* ∎

Theorem 4.2.2. *A constraint defined on a sample T of a relation R applies also on R. On the other hand, a constraint* $\ll\mathcal{A}$**:** $C\gg$ *on R applies on T if the set of attributes of T is a superset of $\mathcal{A}$ and there is at least one tuple t_T, the values of which satisfy the condition C.*

Proof: The image of a tuple $t_T \in T$ has all the attributes (and values) of t_T, and possibly some extra. Thus, any condition satisfied by t_T will also be satisfied by its image, and since all the attributes of t_T are present in its image, any constraint violated by t_T will

[1] This is the way privacy experts perform their task, as resulting from discussions we had with them.

also be violated by its image. On the other hand, if a constraint $\ll\mathcal{A}\colon C\gg$ is violated by a tuple t in R and there is a tuple $t_T{\in}T$ that satisfies the condition C, since the attributes of T are a superset of $\mathcal{A}$, t_T will also violate the constraint. ■

A graphical illustration of a relation R and of the constraints that are also applying on a sample of it can be found in Figure 4.1. The sample tuples and their images can be identified in the figure by the same tuple number indicated next to the table.

Example 4.2.5. *Table 4.4a shows a view with the diagnosis by age class and place of birth for women. The privacy violating values are masked with a '*'. For example the names of the patients victim of raping are masked. Notice how the second tuple is not masked because the privacy constraint $\ll DOB, POB\colon$ Diagnosis = Sclerosis$\gg$ does not apply to the view since the field DOB is not shown.* ■

Example 4.2.6. *Table 4.4b shows a view with the diagnosis by age class, place and date of birth for men. The tuples of patients affected by STD do not show the actual values of names, age class and Place of Birth. Similarly the date and place of birth of the last tuple is hidden as this time the constraint $\ll DOB, POB\colon$ Diagnosis = Sclerosis$\gg$ applies to the view.* ■

4.3 Problem

We are facing two different problems that we are called to solve: the *privacy elicitation* and the *constraint specification* problems. In the following we provide a formalization of such problems and their solutions.

4.3.1 Privacy Constraints elicitation

The *privacy elicitation* problem deals with the collection of the constraints from the privacy expert. Assume the existence of a set of privacy constraints P on a relation R, each one having an attribute set at most k. Without taking into consideration the set P, and assuming a display limitation of N attributes and M tuples for a relation, with $N{\leq}k$, the *privacy elicitation problem* requires the discovery of an ordered list of samples of size NxM that maximizes the likelihood that the privacy expert will specify the set P as soon as possible.

An exhaustive approach is to create all the possible samples from R and then select the ones which are more likely to contain privacy constraints.

Intuitively, the more representative are the values in a sample the more likely it will contain a violation. This because values occurring together very frequently in the relation can originate many privacy violations. In constructing the sample we will exploit this fact by assembling the sample giving preference to frequently occurring values. In order to do that we use the notion of pattern and frequent pattern defined below.

Definition 4.3.1. *A pattern in a relation R is a conjunction of primitive conditions of the form $\mathcal{C} = \{(A_1{=}v_1) \wedge \ldots \wedge (A_k{=}v_k)\}$ selecting tuples in R. We indicate with $R(\mathcal{C}) = \{t \in R : t \text{ satisfies } \mathcal{C}\}$ the tuples covered by the pattern and with $|R(\mathcal{C})|$ the pattern* frequency*.* ■

In data mining values occurring together frequently (more than a certain threshold) in a data set are defined *frequent patterns* [82].
The more frequently occurring combinations of primitive conditions correspond to the more representative values in the table that are capable to cover many tuples of the relation. By showing such representative values in a sample the user can potentially notify an high number of violations looking just at few tuples.

Definition 4.3.2. *Given a pattern $\mathcal{C} = \{(A_1{=}v_1) \wedge \ldots \wedge (A_k{=}v_k)\}$ on R with frequency $|R(\mathcal{C})| = k$ can be generated $2^{|C|} * k$ constraints.* ■

A pattern partitions the relation R into sets of tuples satisfying certain conditions and projected on some attributes. From Definition 4.3.2 as the size of the samples $|C|$ is fixed in order to maximize the number of constraints definable with a pattern we try to maximize the frequency k.
In doing that we follow a greedy approach by constructing the sample tuple by tuple starting from the most frequent patterns. Patterns represented by tuples already presented to the user in a sample can be excluded from the samples that follow. In this way, each sample contains always at least a new combination of values.
By considering the patterns with higher frequency we basically give higher priority to the more common values in the table. The sampling process ends when either there are no more tuples in R to show to the user or a suitable level of coverage of R by means of samples is reached.

Many algorithms are available in literature to compute efficiently frequent patterns like the FP-growth methodology [81] and open source implementations are also available (e.g. Apache Mahout[2] [103]). FP-growth is based on the frequent-pattern tree (FP-Tree) structure created from the list of transactions where each transaction is a set of items. An FP-Tree provides a compact representation of a set of transactions and allows to get

[2] Apache Mahout,Parallel Frequent Pattern Mining
https://cwiki.apache.org/confluence/display/MAHOUT/Parallel+Frequent+Pattern+Mining

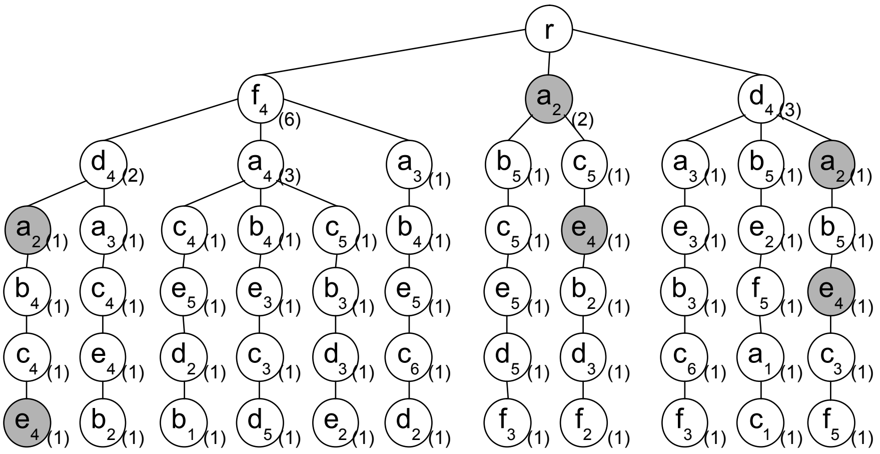
the frequency of all the patterns contained in the input transactions. The goal of the FP-growth algorithm is to explore efficiently the FP-Tree to find the set of items that are more frequently occurring together, that is the frequent patterns.
In our case, a transaction is a tuple in the relation R and the items are the primitive conditions in the where clause of the query selecting the tuple. For example, tuple 1 in Table 4.1 is selected by the query

`select` $Name$, $AgeClass$, DOB, POB, Sex, $Department$, $Symptom$, $Diagnosis$
`from` R
`where` $Name =$'$Sophy$' `and` $AgeClass =$'$18 - 30$'
  `and` $DOB =$'02/10/1983' `and` $POB =$'$Meano$'
  `and` $Sex =$'F'`and` $Department =$'$Physiotherapy$'
  `and` $Symptom =$'$Paralysis$'`and` $Diagnosis =$'$Sclerosis$'

and corresponds to the transaction ($Name$ = '$Sophy$', $AgeClass$ = '18 − 30', DOB = '02/10/1983', POB = '$Meano$', Sex = 'F', $Department$ = '$Physiotherapy$', $Symptom$ = '$Paralysis$', $Diagnosis$ = '$Sclerosis$).

	A	B	C	D	E	F
1	a_2	b_4	c_4	d_4	e_4	f_4
2	a_2	b_5	c_5	d_5	e_5	f_3
3	a_3	b_3	c_6	d_4	e_3	f_3
4	a_2	b_2	c_5	d_3	e_4	f_2
5	a_4	b_1	c_4	d_2	e_5	f_4
6	a_4	b_4	c_3	d_5	e_3	f_4
7	a_1	b_5	c_1	d_4	e_2	f_5
8	a_3	b_2	c_4	d_4	e_2	f_4
9	a_4	b_3	c_5	d_3	e_2	f_4
10	a_3	b_4	c_6	d_2	e_5	f_4
11	a_2	b_5	c_3	d_4	e_4	f_5

(a) Table R



(b) FP-Tree of sample Table R

Figure 4.2: Example of FP-Tree of DB table.

An FP-Tree provides a concise representation of the relation and allows to explore it to derive the frequent patterns. For example the FP-Tree for the table in Figure 4.2a looks like the one depicted in Figure 4.2b. The number in brackets is the support of the path from the root node (r) to the labelled node in the specific branch of the tree. The frequency of an atomic condition at a node is the sum of the support of all the nodes in which the condition occurs. For example node f_4, representing the single condition pattern $F = f_4$ in table R, has frequency 6. For more complex patterns given by the conjunction of different atomic conditions corresponding to a path in the tree, the frequency is given by the sum of the support of the less supported nodes in each path in which the pattern occurs. For example, the pattern with conditions $A = a_2 \wedge E = e_4$ has frequency 3 that is given by the sum of the lower supported nodes of the pattern containing both $A = a_2$ and $E = e_4$ as highlighted in Figure 4.2 (that in all the 3 cases is equal to 1).

Definition 4.3.3. *Given a relation R with FP-Tree T and root r, we define support of a node n "support(n)" as the number of paths from the leaf nodes to node n.* ■

A frequent pattern gives us the combinations of primitive conditions that are more occurring in the table. For example, the more occurring pattern in Table 4.1 is $Sex =$ 'F'.

A solution based on the exhaustive enumeration of all the possible patterns in the sampled table R guarantees that all the value combinations in R are considered as it generates all the possible constraints in R. Consequently by Definition 4.3.2 no privacy constraints will be lost. However, it is impractical to enumerate all the possible patterns as they are exponential in the number of attributes.

Definition 4.3.4. *A relation R with N attributes and m rows produces at most $(2^m - 1)\binom{N}{k}$ samples with k attributes. The maximum number of patterns in R is $\sum_{i=1}^{N}(2^m - 1)\binom{N}{i} = (2^m - 1)(2^N - 1)$.* ■

It is common in frequent pattern mining algorithms to prune the patterns with a frequency below a certain minimum threshold (minimum support). In our scenario we cannot apply the same solution as in this way we will loose patterns on which the user may want to define constraints.

Instead, we propose an heuristic to reduce the number of patterns to consider but that can still show all the tuples in R. However, the intent is to avoid showing all the tuples in R but instead to cover as much privacy constraints as possible of those the user has in mind with a limited number of samples.

Sample Generation

The approach is depicted in Figure 4.3. The intuition is to choose the tuples which are more likely to contain privacy violations and that allow to cover a big region of R that is the tuples which expose more values of R. This tends to reduce the number of samples required to cover R and chooses first the samples which are more likely to contain privacy violations. We do not consider all the possible frequent patterns in table R but we defined a policy to choose the tuples in the sample and then another policy to choose the set of attributes on which to project. The idea is to give more importance to the tuples exposing more values of R because by showing them to the user we expose a wide region of R. In order to do that we define the concept of *benefit*.

Definition 4.3.5. *Given a relation R, its FP-Tree T with root node r and a path $\mathcal{C} = \{n_1, n_2, \ldots, n_k\}$ in T, such that $n_i = (A_i{=}v_i)$ for $i = 1, \ldots, k$ is the node at level i of the FP-Tree, we define benefit of a node n_i, $\mathcal{F}(n_i) = (n_i)$, as:*

1. *if the parent of node n_i is r then $\mathcal{F}(n_i) = support(n_i)$*
2. *else $\mathcal{F}(n_i) = \mathcal{F}(n_{i-1}) + support(n_i)$*

■

Intuitively the benefit measure how many values a path can cover and is obtained by summing up the node support with the support of each ancestors of the node in the FP-Tree. The intuition is graphically represented in Figure 4.3 with the grey cells indicating the values in the tuples each prefix of the path on the relation's attribute selects. For example let the path be $\mathcal{C} = \{(A = a_2) \wedge (B = b_4) \wedge (C = c_4) \wedge (D = d_4) \wedge (E = e_4) \wedge (F = f_4)\}$ which corresponds to the first tuple in Table 4.2a: it has benefit 13 which is also the higher benefit in that FP-Tree.

In this way even nodes with a low frequency in a certain path like $E = e_4$ will gain importance because they bring with them other nodes in the path that are occurring frequently. Notice that a path from the root node to the leaf of the FP-Tree gives a list of atomic conditions identifying a single tuple which exists in the table.

From the considerations above the leaf nodes with higher benefit select tuples with the more frequently occurring values in a certain path. The approach can be iterated as follows: i) choose the tuple with higher benefit to be included in the sample and remove it from R; ii) reconstruct the FP-Tree with the remaining tuples; iii) recompute the benefit; iv) choose the next tuple with higher benefit to be included in the sample.

This approach gives already a sample composed of tuples in R which basically partitions the table in horizontal chunks with total benefit given by the sum of the benefits of its tuples. However, our problem requires samples with a maximum number of attributes by projecting the tuples on a certain set of attributes. The set of attributes on which to project should maximize the coverage of the values in table R. We define a maximal coverage sample as follows:

Definition 4.3.6. *Given a relation R with attributes $\mathcal{A} = \{A_1, \ldots, A_k\}$ and n tuples and a set of tuples $\mathcal{T} = \{t_1, \ldots, t_M\} \subseteq R$ such that each tuple t_i is given by the query $R(t_i)|_{\mathcal{A}} = \sigma_{C_i|_{\mathcal{A}}}(R)$ with $C_i|_{\mathcal{A}} = \bigwedge_{A_j \in \mathcal{A}}(A_j = t_i[A_j])$ and given $\mathcal{T}_{\mathcal{A}} \subseteq \mathcal{A}$ such that $|\mathcal{T}_{\mathcal{A}}| = N$ a set of N attributes of R, we said $\pi_{\mathcal{T}_{\mathcal{A}}}\big(\sigma_{\bigvee_{t_i \in \mathcal{T}}(C_i|_{\mathcal{T}_{\mathcal{A}}})}(R)\big)$ is a maximal coverage sample if $\nexists \mathcal{A}' \subseteq \mathcal{A}$ such that $\mathcal{A}' \neq \mathcal{T}_A$ and $|Img(\sigma_{\bigvee_{t_i \in \mathcal{T}}(C_i|_{\mathcal{T}_{\mathcal{A}}})}(R))| < |Img(\sigma_{\bigvee_{t_i \in \mathcal{T}}(C_i|_{\mathcal{A}'})}(R))|$.* ■

Notice we are using the relational algebra notation [24] to express the query returning the tuples in $\mathcal{T}$ using only the attributes in $\mathcal{T}_A$ and $\mathcal{A}'$.

The definition above states that given a set of tuples with maximum benefit, a maximal coverage sample is a projection of these tuples on a set of attributes giving the larger image on R (see the projected area in Figure 4.3). In order to find an optimal set of attributes maximizing the size of the image, all the combinations of attributes should be tested.

In the next section is given an algorithm for the creation of samples satisfying the definitions of benefit and maximal coverage given above to repeatedly produce samples from a relation.

4.3.2 Privacy Constraints Specification

The *constraint specification* problem deals with deriving a descriptive and concise definition of privacy constraints from the violations indicated by the users on the samples. The conditions identifying a single tuple indicated by the user as violating produce a constraint that is typically too specific as it identifies only a specific tuple and cannot cover other privacy violations.

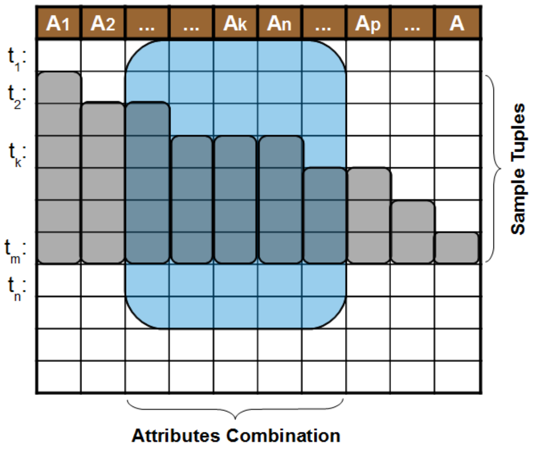


Figure 4.3: The sampling approach selects first the tuples with higher benefit and then the attributes giving higher coverage.

By removing some conditions it is possible to obtain a more general constraint with a shorter list of conditions which covers more tuples and consequently also more privacy violations compared to a single tuple constraint. In this way the overall number of constraints required to represent all the violations identified in the relation R can be reduced. In the following it is defined when a privacy constraint should be considered valid with respect to the indications given by the privacy experts.

Definition 4.3.7. *Given a set of privacy constraints* $\mathbb{C} = \{\ll\mathcal{A}\mathbf{:}\, C\gg\}$ *on* R defined by the user *to hide the same set of attributes* $\mathcal{A}$ *and given a privacy constraint* $p = \ll \mathcal{A}_p : C_p \gg$ *in* R *such that* $\mathcal{A}_p = \mathcal{A}$*,* p *is a valid privacy constraint,* $\mathbb{C} \Longrightarrow_v^R p$*, if* $\widehat{R}(p) \subseteq \widehat{R}(\mathbb{C})$*.* ∎

Intuitively, definition 4.3.7 says that a privacy constraint is valid if it does not select more tuples than what the user has indicated. An invalid privacy constraint for relation R corresponds to a privacy constraint that is too much generic meaning that it will select also tuples that are perfectly fine for the user on that set of attributes.
Notice how the validity property is defined only for privacy constraints selecting the same set of attributes to hide. This because it does not make sense to compare constraints hiding different attributes as they are not comparable (e.g. $\ll Name\mathbf{:}POB =$'*Trento*'$\gg$ cannot be compared with $\ll POB\mathbf{:}Name =$'*Anna*'$\gg$). In addition, all the constraints are defined on the same relation R.
Our goal is to find a set of constraints that allows to identify all and only the privacy violations defined by the user. A set of privacy constraints is a *solution* for the set of constraints given by the user when the following definition is satisfied:

Definition 4.3.8. *Given a set of constraints* $\mathbb{C} = \{\ll\mathcal{A}\mathbf{:}\, C\gg\}$ *on* R defined by the user *to hide the same set of attributes* $\mathcal{A}$ *and given a set of privacy constraint* $P = \{\ll \mathcal{A}_p : C_p \gg\}$ *in* R *such that* $\mathcal{A}_p = \mathcal{A}$*,* P *is said to be a valid solution for* $\mathbb{C}$ *in* R*, i.e.* $P \in \mathbb{S}(\mathbb{C})$*, if* $\forall p \in P, \mathbb{C} \Longrightarrow_v^R p$ *and* $\widehat{R}(P) = \widehat{R}(\mathbb{C})$*.* $\mathbb{S}(\mathbb{C})$ *is the set of all the valid solutions for* $\mathbb{C}$*.* ∎

Given the definitions above the constraints specification problem can be formulated as follows: given the collection of *candidate privacy constraints* specified by the privacy violations indications given by the privacy experts, we would like to derive a valid set of non redundant privacy constraints with the minimum number of conditions.
More formally a *minimal valid solution* can be defined as:

Definition 4.3.9. *Given a set of constraints* $\mathbb{C} = \{\ll\mathcal{A}\mathbf{:}\, C\gg\}$ *on* R defined by the user *to hide the same set of attributes* $\mathcal{A}$ *and given a set of privacy constraint* $P = \{p_1, \ldots, p_n\}$ *in* R*,* P *is said to be a minimal valid solution for* $\mathbb{C}$ *if* $P \in \mathbb{S}(\mathbb{C})$ *and* $\nexists Q \in \mathbb{S}(\mathbb{C})$ *such that* $\sum_{\ll\mathcal{A}:C\gg\in\mathcal{Q}} |(\|C\|)| < \sum_{\ll\mathcal{A}:C\gg\in\mathcal{P}} |(\|C\|)|$*.* ∎

Intuitively, a set of valid privacy constraints represents a minimal valid solution for a set of privacy constraints derived from the violations indicated by the user, if there is no other valid solution for the given set of privacy constraints using less conditions.

The following theorem gives a mechanism to reduce the number of constraints in a valid set of privacy constraints still maintaining its validity.

Theorem 4.3.1. *Given a set of constraints* $\mathbb{C} = \{\ll\mathcal{A}\textbf{:}\ C\gg\}$ *defined by the user in* R *hiding the same set of attributes* $\mathcal{A} = \{A_1, \ldots, A_k\}$ *such that* $\mathbb{S}(\mathbb{C})$ *is the set of solutions for* $\mathbb{C}$ *and let* $p' = \ll\mathcal{A}\textbf{:}\ c'\gg$ *be a valid privacy constraint in* R *hiding the attributes* $\mathcal{A}$ *such that* $\mathbb{C} \Longrightarrow_v^R \ll\mathcal{A}\textbf{:}\ c'\gg$. *Then* $\mathbb{C}' = p' \cup (\mathbb{C} \setminus \{p \in \mathbb{C} | p \dot{\leq} p'\}) \in \mathbb{S}(\mathbb{C})$. ∎

Proof To prove $\mathbb{C}'$ is a valid solution it should be verified that $\widehat{R}(\mathbb{C}') = \widehat{R}(\mathbb{C})$ which can be proved in two steps: (i) $\widehat{R}(\mathbb{C}') \subseteq \widehat{R}(\mathbb{C})$ (ii) $\widehat{R}(\mathbb{C}) \subseteq \widehat{R}(\mathbb{C}')$.

(i) By construction $\mathbb{C}'$ contains only valid constraints which implies $\widehat{R}(\mathbb{C}') \subseteq \widehat{R}(\mathbb{C})$.

(ii) Given that $\mathbb{C}' = p' \cup (\mathbb{C} \setminus \{p \in \mathbb{C} | p \dot{\leq} p'\})$ it is enough to show that $\widehat{R}(\{p = \ll\mathcal{A}\textbf{:}\ c\gg \in \mathbb{C} | p \dot{\leq} p'\}) \subseteq \widehat{R}(p')$. From the definition of subsumption $\forall p = \ll\mathcal{A}\textbf{:}\ c\gg, p \in \{p \in \mathbb{C} | p \dot{\leq} p'\}$, $p \dot{\leq} p'$ implies that $c \Rightarrow c'$ that is $\widehat{R}(p) \subseteq \widehat{R}(p')$ which implies $\widehat{R}(\{p = \ll\mathcal{A}\textbf{:}\ c\gg \in \mathbb{C} | p \dot{\leq} p'\}) \subseteq \widehat{R}(p')$ and consequently $\widehat{R}(\mathbb{C}) \subseteq \widehat{R}(\mathbb{C}')$.

From (i) and (ii) we got $\widehat{R}(\mathbb{C}') = \widehat{R}(\mathbb{C})$ and this proves that $\mathbb{C}'$ is a valid solution. ∎

Intuitively, a group of privacy constraints hiding the same set of attributes can be combined in a more general constraint if the resulting constraint selects the same set of tuples meaning that it does not select more tuples than what the user has indicated as violating. Basically, given a valid solution if we substitute with a valid privacy constraints p all the constraints that are subsumed by p the validity of the solution is not compromised.

A brute force approach to find the minimal valid solution is to consider all the possible combinations of constraints that can be defined from the violations notified by the user, and to generalize them by considering the common conditions among the constraints until a valid constraint using less conditions is found. In the worst case this corresponds to the original *candidate privacy constraints*. This approach guarantees to find a global optimum solution but it is infeasible as it considers a prohibitive number of combinations. For example with c constraints on α attributes the upper bound of the number of privacy constraints combinations to check is given by: $\big(2^\alpha - 1\big)\big(2^c - 1\big)$.
In order to avoid dealing with such an exponential number of cases we adopt a greedy approach. Theorem 4.3.1 illustrates a procedure to reduce the size of a solution by properly choosing a valid constraint to eliminate all the subsumed constraints from the

solution. Our greedy approach chooses the valid privacy constraint p subsuming the maximum number of privacy constraints as described below.

Privacy Constraints Minimization

Assume a set of valid privacy constraint $\mathbb{C}$ defined by the user to hide the same set of attributes is given and an optimal valid solution $\mathbb{P} \in \mathbb{S}(\mathbb{C})$ with less conditions than $\mathbb{C}$ should be found. For each privacy constraint $p = \ll \mathcal{A}\text{:}C \gg$, $p \in \mathbb{S}(\mathbb{C})$ with $K = \|C_p\|$ atomic conditions, there may be $2^{|K|}-1$ generalizations of the constraint p, as many as the subsets of conditions in $\|C_p\|$. For example the constraint $\ll \mathcal{A}\text{:}(A = a_1 \wedge B = b_1 \wedge C = c_1) \gg$ has 7 generalizations: $\ll \mathcal{A}\text{:}(A = a_1) \gg$, $\ll \mathcal{A}\text{:}(B = b_1) \gg$, $\ll \mathcal{A}\text{:}(C = c_1) \gg$, $\ll \mathcal{A}\text{:}(A = a_1) \wedge (B = b_1) \gg$, $\ll \mathcal{A}\text{:}(A = a_1) \wedge (C = c_1) \gg$, $\ll \mathcal{A}\text{:}(B = b_1) \wedge (C = c_1) \gg$ and the constraint itself $\ll \mathcal{A}\text{:}(A = a_1 \wedge B = b_1 \wedge C = c_1) \gg$.

Each generalization covers a certain number of privacy constraints in $\mathbb{C}$ not yet covered by the optimal solution $\mathbb{P}$. Let $\mathbb{G} = \{g_1, \ldots, g_n\}$ be the set of all the possible generalizations of privacy constraints in $\mathbb{C}$.

The greedy approach finds the generalization $g = \ll \mathcal{A}\text{:}C_g \gg \in \mathbb{G}$ such that it covers the higher number of privacy constraints in $\mathbb{C}$ not yet covered in $\mathbb{P}$ and with the minimum number of conditions.

If g is valid it is added to the solution $\mathbb{P}$ and by Theorem 4.3.1 all the privacy constraints subsumed by g can be removed from $\mathbb{G}$ because they are covered by the solution $\mathbb{P}$.

If g has only a partial overlapping with a constraint $q \in \mathbb{G}$, $q = \ll \mathcal{A}\text{:}C_q \gg$ meaning that $\widehat{R}(q) \nsubseteq \widehat{R}(g)$ then q cannot be substituted with g in $\mathbb{P}$. However, the capability of q to hide tuples not yet covered in $\mathbb{P}$ is reduced as part of the tuples it requires to mask are already masked by g. This means that the coverage of q should be reduced by the tuples occurring in the overlapping with g. This corresponds to update the coverage of q to $|\widehat{R}(q)| - |\widehat{R}(q) \cap \widehat{R}(g)|$ which is equivalent to compute $|\widehat{R}(\ll \mathcal{A}\text{:}C_q \wedge \overline{C_g} \gg)|$.

The update of the coverage guarantees to choose always the generalization covering the higher number of privacy constraints in $\mathbb{C}$ not yet covered by $\mathbb{P}$.

The FP-Tree structure can be used to compute efficiently the set of generalizations $\mathbb{G}$ as it allows to generate all the possible combinations of conditions shared by different privacy constraints in $\mathbb{C}$. Such combinations corresponds to the frequent patterns of the tuples in $\widehat{R}(\mathbb{C})$ as defined in Section 4.3.1 ordered in decreasing order of coverage and increasing length (patterns with less atomic conditions come first). As said above each time the first valid solution g in the ordered list of frequent patterns is selected to be added to the valid solution set $\mathbb{P}$, the list of frequent patterns is updated removing all the patterns subsumed by g and reordered to reflect the change in coverage of the patterns partially covered by the solution. In order to do that, the complement of the solution is added to the remaining patterns and their coverage is recomputed. Actually, the complementary patterns serve the sole purpose of generating the coverage of the patterns considering only tuples not yet covered by a privacy constraint in the solution.

As shown in Theorem 4.3.2 adding the complement of a valid constraint q to a privacy constraint p gives a constraint with the same validity property of p (according to the validity definition in Definition 4.3.8). Intuitively Theorem 4.3.2 shows that adding to an invalid pattern the negated conditions of a valid pattern will not transform the pattern in a valid pattern. Similarly, adding to a valid pattern the negated condition of a valid pattern will not transform the pattern in an invalid pattern. This implies it is not necessary to include the negated conditions in the definition of the privacy constraints which results from the minimization process described above but the solution will be constructed considering only the "positive" constraints.

Theorem 4.3.2. *Given the privacy constraints $p = \ll\mathcal{A}\text{:}\ C_p\gg$ and $q = \ll\mathcal{A}\text{:}\ C_q\gg$ in a relation R and given $\mathbb{C} = \{\ll\mathcal{A}\text{:}\ c\gg\}$ set of valid privacy constraints for R with $q \in \mathbb{C}$ valid privacy constraint then p is valid iff $\ll \mathcal{A} : C_p \wedge \overline{C_q} \gg$ is valid.* ■

Proof The proof can be split in four cases: (i) if p is valid in R then $\ll \mathcal{A} : C_p \wedge \overline{C_q} \gg$ is valid in R; (ii) if $\ll \mathcal{A} : C_p \wedge \overline{C_q} \gg$ is valid in R then p is valid in R; (iii) if $\ll \mathcal{A} : C_p \wedge \overline{C_q} \gg$ is not valid in R then p is not valid in R; (iv) if p is not valid in R then $\ll \mathcal{A} : C_p \wedge \overline{C_q} \gg$ is not valid in R.

(i) Given p and q valid in R then we can say $\widehat{R}(q) \subseteq \widehat{R}(\mathbb{C})$ and $\widehat{R}(p) \subseteq \widehat{R}(\mathbb{C})$. By the set properties we can say that $\widehat{R}(\ll \mathcal{A}\text{:}C_p \wedge \overline{C_q} \gg) = \widehat{R}(p) \setminus \widehat{R}(q) \subseteq \widehat{R}(\mathbb{C})$ and consequently $\ll \mathcal{A}\text{:}C_p \wedge \overline{C_q} \gg$ is valid, $\mathbb{C} \Longrightarrow_v^R \ll \mathcal{A}\text{:}C_p \wedge \overline{C_q} \gg$

(ii) Given $\ll \mathcal{A}\text{:}C_p \wedge \overline{C_q} \gg$ and q valid in R then $\widehat{R}(\ll \mathcal{A}\text{:}C_p \wedge \overline{C_q} \gg) \subseteq \widehat{R}(\mathbb{C})$. By the

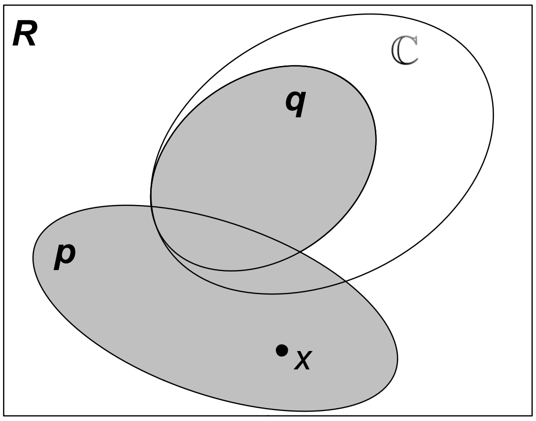


Figure 4.4: Overlapping valid and invalid patterns.

set operators properties $\widehat{R}(\ll \mathcal{A}{:}C_p \gg) = \widehat{R}(\ll \mathcal{A}{:}C_p \wedge \overline{C_q} \gg) \cup \widehat{R}(\ll \mathcal{A}{:}C_p \wedge C_q \gg)$. By hypothesis each term of the union is $\subseteq \widehat{R}(\mathbb{C})$ which implies $\widehat{R}(p) \subseteq \widehat{R}(\mathbb{C})$ that is p is valid, $\mathbb{C} \Longrightarrow_v^R p$.

(iii) Given $\ll \mathcal{A} : C_p \wedge \overline{C_q} \gg$ not valid and q valid in R. From hypothesis $\widehat{R}(\ll \mathcal{A}{:}C_p \wedge \overline{C_q} \gg) \nsubseteq R(\mathbb{C})$ meaning that $\exists x \in \widehat{R}(p)$ such that $x \in \widehat{R}(p) \wedge x \notin \widehat{R}(\mathbb{C})$ which implies that p is not valid.

(iv) Given p not valid (see Figure 4.4) it means $\exists x \in \widehat{R}(p)$ s.t. $x \notin \widehat{R}(\mathbb{C})$ which implies it is also not in $\widehat{R}(q)$. From $x \in \widehat{R}(p) \wedge x \notin \widehat{R}(q)$ we derive $x \in (\widehat{R}(p) \setminus \widehat{R}(q)) \wedge x \notin \widehat{R}(\mathbb{C})$ which means $\widehat{R}(\ll \mathcal{A}{:}C_p \wedge \overline{C_q} \gg) \nsubseteq \widehat{R}(\mathbb{C})$ and so $\ll \mathcal{A} : C_p \wedge \overline{C_q} \gg$ is not valid.

∎

The approach presented in this section finds a minimal valid solution for a set of privacy constraints $\mathbb{C}$ hiding the same set of attributes $\mathcal{A}$. When heterogeneous constraints hiding different attributes should be minimized the same approach can be applied on each single group of homogeneous constraints. The optimal solution for the whole set of constraints is given by the union of the partial optimal solutions discovered for each group. This because each group is independent from the others.
Notice also that the greedy approach is an heuristic which gives in general a solution close to the optimum but is not said to be the global optimum. This is the price to pay to make the algorithm compute the solutions in a reasonable time.

4.4 Algorithm

In this section we present the algorithms to address the privacy constraints elicitation and specification problems:

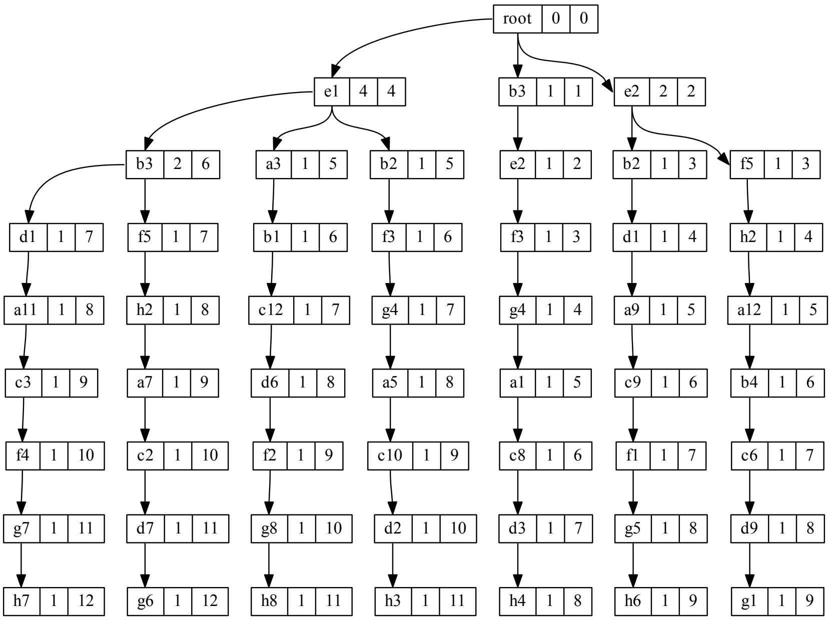
- *Sample Generation*: generates the samples based on the data distribution in the input relation;
- *Privacy Constraints Minimization*: derives a descriptive and concise definition of privacy constraints from the violations indicated by the user.

4.4.1 Sample Generation

Algorithm 2, *getSample*, takes a relation R with n tuples and k attributes ($k*n$ relation) and finds a sample with M tuples and N attributes (NxM relation) chosen from R according to Definition 4.3.5 of benefit and Definition 4.3.6 of maximal coverage sample. It creates the FP-Tree of relation R (line 7) to compute the benefit of each path from the

	A	B	C	D	E	F	G	H
1	a_{11}	b_3	c_3	d_1	e_1	f_4	g_7	h_7
2	a_4	b_4	c_1	d_5	e_2	f_4	g_2	h_7
3	a_3	b_1	c_{12}	d_6	e_1	f_2	g_8	h_8
4	a_6	b_1	c_{11}	d_5	e_1	f_1	g_5	h_5
5	a_{10}	b_4	c_5	d_3	e_1	f_2	g_8	h_8
6	a_1	b_3	c_8	d_3	e_2	f_3	g_4	h_4
7	a_9	b_2	c_9	d_1	e_2	f_1	g_5	h_6
8	a_2	b_4	c_{13}	d_4	e_1	f_5	g_6	h_1
9	a_7	b_3	c_2	d_7	e_1	f_5	g_6	h_2
10	a_4	b_4	c_7	d_8	e_2	f_5	g_5	h_2
11	a_{12}	b_4	c_6	d_9	e_2	f_5	g_1	h_2
12	a_8	b_3	c_4	d_5	e_1	f_3	g_3	h_3
13	a_5	b_2	c_{10}	d_2	e_1	f_3	g_4	h_3

(a) R: from mapping on Table 4.1



(b) FPTree of R after 7 tuples with higher benefit are removed.

Figure 4.5: Example of sampling of Table 4.1.

root node (line 8). The FP-Tree for the relation in Table 4.5a is shown in Figure B.1 in Appendix B. Notice how the values in that table are derived from the example in Table 4.1 by mapping each cell value with a conventional unique value (e.g. *Sophy* is mapped to a_{11}).
The algorithm chooses the leaf node with the highest benefit (line 9) corresponding to the tuple with higher benefit, it adds that tuple to the set of tuples in the sample S_T (line 10) and removes it from R (line 11) as it will be covered by a sample.
The steps (line 6– 12) are repeated until M tuples of the sample are selected or there are no more tuples to be covered in R. At each iteration the FP-Tree is created again with the remaining tuples as its shape and benefit at the nodes will change. This guarantees that each sample contains different tuples and that there are no common tuples in different samples.
The FP-Tree obtained for the example in Table 4.5a after 7 iterations (7 tuples are added to the sample) is shown in Figure 4.5b. Notice how the shape and the values of support and benefit are changed compared to the initial FP-Tree (see Figure B.1 in Appendix B).

The second part of the algorithm is devoted to identify the set of attributes on which to project in order to obtain a *maximal coverage sample* as defined in Definition 4.3.6. We use an exhaustive approach by generating all the possible combinations of N attributes over

Algorithm 2: *getSample*$(R, N, M) \mapsto S$.

```
   Data: $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$, $R \neq \Phi$,
         $R = \{t_1, \ldots, t_n : t_i = [A_1{:}v_1, A_2{:}v_2, \ldots, A_k{:}v_k], i = 1, \ldots, n\}$
   Result: Query $S$ of a sample on $R$ with at most $M$ tuples and $N$ attributes
 1 Let $S_T$ set of tuples of sample $S$
 2 Let $S_A$ set of attributes of sample $S$
 3 $S_T \leftarrow \Phi$
 4 $S_A \leftarrow \Phi$
 5 /* choose M tuples from R                                              */
 6 while ($S_T < M \wedge R \neq \Phi$) do
 7     $T \leftarrow FPTreeCreation(R)$
 8     $computeBenefit(T)$
 9     $t \leftarrow getTopBenefitTuple(\mathbb{T})$
10     $S_T \leftarrow S_T \cup t$
11     $R \leftarrow R \setminus t$
12 end
13 /* choose N attributes from k on which to project the sample tuples     */
14 $C \leftarrow combinations(k, N)$
15 $S_A \leftarrow C[1]$
16 $C \leftarrow C \setminus S_A$
17 while ($C \neq \Phi$) do
18     $\mathcal{A}' \leftarrow C[1]$
19     $C \leftarrow C \setminus \mathcal{A}'$
20     if $\left(\left|Img\left(\sigma_{\bigvee_{t \in S_T}(\bigwedge_{A_i \in S_A}(A_i = t[A_i]))}(R)\right)\right| < \left|Img\left(\sigma_{\bigvee_{t \in S_T}(\bigwedge_{A_i \in \mathcal{A}'}(A_i = t[A_i]))}(R)\right)\right|\right)$ then
21         $S_A \leftarrow \mathcal{A}'$
22     end
23 end
24 return $\pi_{S_A}\left(\sigma_{\bigvee_{t \in S_T}(\bigwedge_{A_i \in S_A}(A_i = t[A_i]))}(R)\right)$
```

the set $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$ (at line 14) which are $\binom{k}{N} = \frac{k!}{N!(k-N)!}$. Each combination is analysed (line 17– 23) to discover the set of attributes S_A that when projected on the sample tuples S_T gives the larger image in R.

The resulting sample is described by the query:

$$S = \pi_{S_A}\left(\sigma_{\bigvee_{t \in S_T}(\bigwedge_{A_i \in S_A}(A_i = t[A_i]))}(R)\right)$$

The sample produced for the table in Figure 4.6a is shown in Figure 4.6b: it has 7 tuples and attributes $\{B, E, F, H\}$.

The next sample to be shown to the user is obtained by applying the same algorithm on the tuples in R not yet covered by a sample. Basically, the tuples in the image of the samples already created are not considered in generating the samples that follow. As the set of tuples in R is finite, the algorithm terminates when all the tuples in R are covered by some sample.

In Appendix B are shown the algorithms for the FP-Tree creation and the frequent patterns discovery (FP-Growth). The implementation follows the algorithm proposed in [82] but it does not perform any pruning to eliminate low frequent items with minimum support below a certain threshold. Alternatively, the procedures described there can be followed exactly providing zero as minimum support parameter.

	A	B	C	D	E	F	G	H
2	a_4	b_4	c_1	d_5	e_2	f_4	g_2	h_7
4	a_6	b_1	c_{11}	d_5	e_1	f_1	g_5	h_5
5	a_{10}	b_4	c_5	d_3	e_1	f_2	g_8	h_8
8	a_2	b_4	$c_1$3	d_4	e_1	f_5	g_6	h_1
9	a_7	b_3	c_2	d_7	e_1	f_5	g_6	h_2
10	a_4	b_4	c_7	d_8	e_2	f_5	g_5	h_2
12	a_8	b_3	c_4	d_5	e_1	f_3	g_3	h_3

(a) Tuples chosen to construct the sample.

	B	E	F	H
2	b_4	e_2	f_4	h_7
4	b_1	e_1	f_1	h_5
5	b_4	e_1	f_2	h_8
8	b_4	e_1	f_5	h_1
9	b_3	e_1	f_5	h_2
10	b_4	e_2	f_5	h_2
12	b_3	e_1	f_3	h_3

(b) Resulting sample: 7 tuples and 4 attributes.

Figure 4.6: Example of sample on Table 4.5a

4.4.2 Privacy Constraints Minimization

Algorithm 3, *MinimizePrivacyConstraints*, takes a relation R and the set $\mathbb{C}$ of candidate privacy constraints defined on R by the user hiding the same set of attributes. It returns a valid solution using the approach described in Section 4.3.2 containing less conditions than the input set $\mathbb{C}$. It may be possible that the procedure is not able to reduce the number of conditions and consequently it will return the same list of constraints provided in input. This depends on the particular set of constraints given in input and on the distribution of the data. For example, a table containing tuples with no values in common among different tuples do not offer any possibility to combine constraints into more general ones. However, it is very unlikely to have totally unrelated values among the tuples of a real database, and it is common to have some degree of overlapping in the data, especially in case of attributes with finite domain.

The algorithm generates the list of frequent patterns (line 1– 2) from the tuples selected

as violating in the input relation R where each tuple is properly transformed into a list of items of the form $Attribute = Value$ representing the conditions selecting the tuple in the relation as defined in Section 4.3.1. The list $\mathbb{G}$ of frequent patterns represents all the possible conditions that can be used to generalize the input constraints. The list is ordered by decreasing order of frequency and increasing length. In this way the more frequent and shortest conditions migrate at the top of the list (line 3).
Until all the privacy constraints in input are covered by the constructed solution (lines 6– 26) the algorithm removes the more frequent pattern g from the top of the list $\mathbb{G}$ (line 7) and if the corresponding constraint is valid ($\widehat{R}(g) \subseteq \widehat{R}(\mathbb{C})$ at line 9) it performs the following actions:

1. add the constraint g to the solution (line 11)
2. remove from $\mathbb{G}$ all the patterns of constraints subsumed by g (lines 12– 16)
3. add the complement of the condition in g to all the patterns in $\bar{\mathbb{G}}$ (lines 18– 22)
4. performs a sorting on the resulting $\bar{\mathbb{G}}$ list removing the patterns with zero frequency (line 23)
5. finally, also the list of frequent pattern in $\mathbb{G}$ used to find the other valid solutions is sorted using the order of their corresponding negated patterns in $\bar{\mathbb{G}}$ (line 24).

Notice how the sampling algorithm and the constraints minimization algorithm are totally independent and the second can work on privacy constraints defined directly on the original huge table as also on the sample views produced by Algorithm 2. Indeed, it is possible to combine more samples together and to apply the minimization algorithm to the resulting combined samples. However, since it is possible to minimize only constraints defined on the same set of attributes (that is the union of the attributes to hide with the attributes in the list of conditions), only samples projected on the same set of attributes can be combined.
Table 4.5 shows the privacy constraints obtained from the minimization algorithm applied on a sample. Notice how each of the 5 candidate privacy constraints derived from the privacy violations in input has 6 conditions corresponding to the attributes in the sample for a total of 30 conditions. The minimization algorithm derives just two constraints with only two conditions.

4.5 Experiments

In this section we present the results of the execution of the sample generation and privacy constraint minimization algorithms on test data. The tests are performed on a relation

Algorithm 3: $MinimizePrivacyConstraints(\mathbb{C}, R) \mapsto \mathbb{P}$

```
Data: $\mathbb{C} = \{\ll\mathcal{A}\textbf{:}\ C\gg\}$ set of candidate privacy constraints on $\mathbb{R}$
Result: $\mathbb{P} = \{\ll\mathcal{A}\textbf{:}\ C\gg\}$ such that $\sum_{\ll\mathcal{A}:\mathcal{C}\gg\in\mathcal{P}} |(\|C\|)| \leq \sum_{\ll\mathcal{A}:\mathcal{C}\gg\in\mathcal{C}} |(\|C\|)|$
$\mathbb{T} \leftarrow FPTreeCreation(\mathcal{C})$ /* creates FP-Tree from violating tuples in $R$ */
$\mathbb{G} \leftarrow FPGrowth(\mathbb{T}, null)$ /* generate frequent patterns from $\mathbb{T}$ */
$sort(\mathbb{G})$ /* sort patterns by decreasing frequency and increasing length */
$\mathbb{P} \leftarrow \Phi$ /* initialize solution set */
$\bar{\mathbb{G}} = \mathbb{G}$ /* initialize ordered list of complementary patterns */
while $(\mathbb{G} \neq \Phi) \wedge (\widehat{R}(\mathbb{P}) \neq \widehat{R}(\mathbb{C}))$ do
    $g \leftarrow \mathbb{G}[1]$ /* take the more frequent and shorter pattern */
    $\mathbb{G} \leftarrow \mathbb{G} \backslash g$
    if $(\widehat{R}(g) \subseteq \widehat{R}(\mathbb{C}))$ then
        /* if the pattern is valid */
        $\mathbb{P} \leftarrow \mathbb{P} \cup g$ /* add the pattern to the solution */
        foreach $(p \in \mathbb{G})$ do
            if $(p \overset{\cdot}{\leq} g)$ then
                /* remove patterns subsumed by $g$ */
                $\mathbb{G} \leftarrow \mathbb{G} \backslash p$
            end
        end
        foreach $(q \in \bar{\mathbb{G}})$ do
            /* add negated condition of $g$ to complementary patterns */
            $\bar{\mathbb{G}} \leftarrow \bar{\mathbb{G}} \backslash q$
            $\bar{\mathbb{G}} \leftarrow \bar{\mathbb{G}} \cup (q \wedge \bar{g})$
        end
        $\mathrm{sort}(\bar{\mathbb{G}})$ /* sort $\bar{\mathbb{G}}$ and remove zero-frequent patterns */
        $\mathrm{sort}(\mathbb{G}, \bar{\mathbb{G}})$ /* sort $\mathbb{G}$ based on frequency in $\bar{\mathbb{G}}$ */
    end
end
return $\mathbb{P}$
```

Table 4.5: Privacy constraints minimization on violations in sample table.

Tuple	Name	AgeClass	DOB	POB	Sex	Department	Privacy Constraint
1	Sophy	$18 - 30$	02/10/1983	Meano	F	Physiotherapy	
3	Helene	< 18	28/01/1995	Trento	F	Psychology	$\ll Name$:$(AgeClass < 18) \gg$
4	Ketty	< 18	13/12/2009	Povo	F	Infectiology	
6	Bob	$18 - 30$	09/05/1991	Daone	M	Oncology	
7	Paul	> 65	10/06/1941	Brenta	M	Infectiology	
8	Chris	$30 - 65$	12/5/1975	Lavis	F	STD	$\ll Name, AgeClass, POB$:$(Department = \text{“}STD''') \gg$
9	Lidia	$18 - 30$	02/08/1982	Vela	F	STD	
10	Jeremy	$30 - 65$	07/03/1976	Vigo	M	STD	

R randomly generated with Gaussian and Poisson distributions. We consider a number of tuples n equal to 100 in the tests of the minimization and 1000 in the tests for the sampling; this because in the former case a sample of more than 100 tuples will be too big for the standard user to deal with. The tuples have $N = 15$ attributes and each attributes has a domain of 10 values.

The test data are generated in such a way that there is no significant overlapping among the tuples as it would be typically in real data. This means that on average our algorithms will perform poorly on this fake data set compared with the same tests on “real” data. We plan to repeat the tests on the real data to evaluate the performances and effectiveness and the different behaviour with a more homogeneous data distribution.

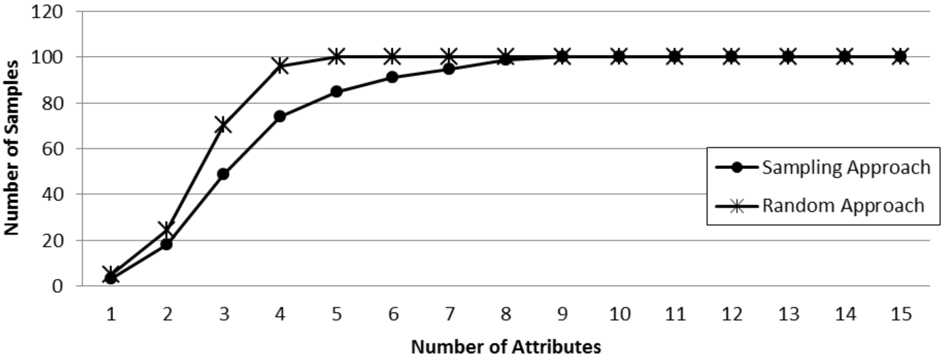
In order to evaluate the sampling algorithm we performed two types of tests: *relation coverage test*, to measure how good the samples are in covering the relation R with different number of sample attributes; *constraints coverage test*, to measure the capability of the samples in capturing the privacy constraints with different number of samples attributes. Both tests are compared to a random sampling algorithm generating the samples with a random choice of the tuples and attributes on which to project. In addition, the number of tuples in the sample is fixed to $M = 10$ while the number of attributes ranges from $N = 1$ to 15. We think that 15 attributes are a reasonable big number of attributes in a sample to propose to a user. An higher number of attributes would risk to be unreadable [83].

In Figure 4.7 we show how many samples should be proposed to the user in order to cover all the relation R. On average we notice 13% less samples are required to cover the whole relation R compared to the random approach selecting both tuples and attributes randomly.

It is interesting to note that with just 1 attribute few interactions are required to cover the whole input relation (3 steps are enough). This because the sampling strategy will basically choose the attribute with higher frequency in the set of tuples given by the tuples selection phase and this will cover for sure a big chunk of relation R. However, a

single attribute sample is quite useless as it does not show to the user a rich set of data on which to define the privacy constraints. The only kind of privacy constraints definable on a single attribute sample are constraints using just this attribute. As the number of attributes increases the algorithm will reach the coverage after an higher number of interactions. This because each attribute added will make the sample more specific and representative of less tuples in R (that is it will have a smaller image). This increases the number of samples to be shown to the user but guarantees to cover much more values (and most importantly combinations of values).
At the other extreme showing to the user a sample of the same number of attributes of relation R is not giving any advantage as it is basically dividing R into chunks of $M = 10$ tuples. in this case the coverage will be reached after $|R|/M$ samples. These results can be used to decide the most suitable number of attributes to be used in the samples given a certain data distribution.

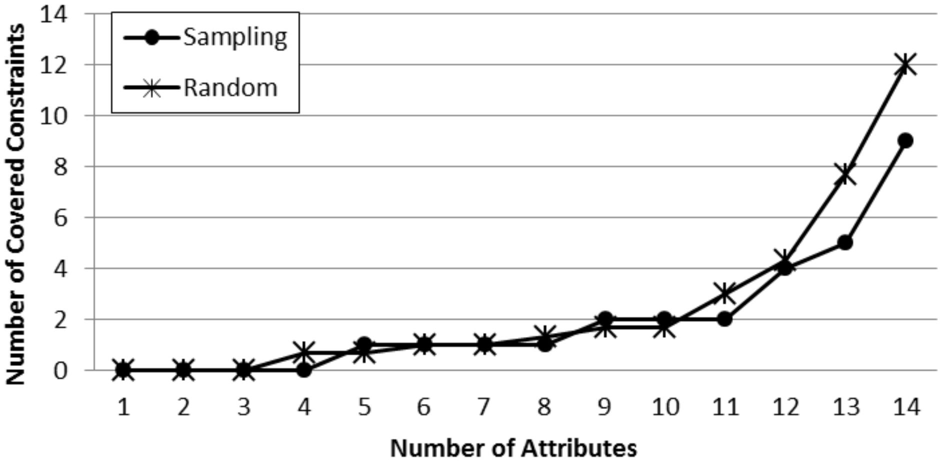
Figure 4.7: Relation coverage test with samples of 10 tuples and at most 15 attributes, relation R with 15 attributes and 1000 tuples.



In Figure 4.8 we show how good the samples are in capturing privacy constraints the user may want to express. In order to perform this test, we generate a random set of $P = 30$ constraints on 15 attributes choosing also randomly the number of attributes to hide and the conditions on which the violations occur. We used a relation R distributed with a *Poisson* distribution to evaluate how good is our sampling strategy to exploit the data distribution. Then we create samples of 10 tuples varying in each trial the number of attributes per sample from 1 up to 14 (we omit 15 as it gives samples with the same dimension of R that will surely cover all the constraints).

In each test trial we check how many constraints can be defined by the user looking at the samples generated.
The sampling approach does not provide such a great improvement on the number of

Figure 4.8: Relation coverage test with samples of 10 tuples and at most 15 attributes, relation R with Poisson distribution ($\lambda = 5$) on 15 attributes and 1000 tuples.

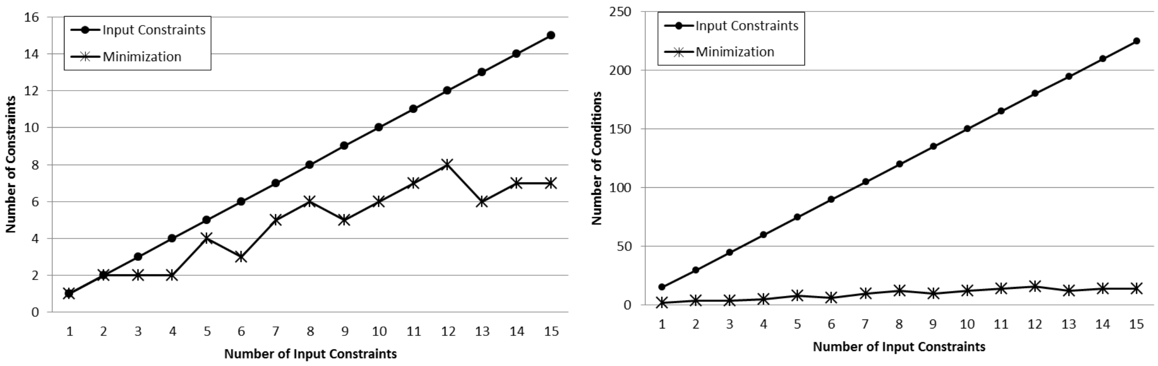


constraints captured, but given that it requires less samples to finish it is not a bad result. For samples of medium size (from 5 to 10 attributes) our sampling is performing at least as good as the random approach. We should also say that the distribution of the data and of the sample generated in these tests are not reflecting the behaviour of the user. In this configuration it is reasonable that a random try can success to guess some of the randomly generated constraints. In order to better evaluate and tune our solution we need to simulate with lab test our approach using a more realistic model of the user behaviour. In addition, a final test with actual data and real users is also needed and we are planning to do it as future work.

In order to evaluate the capability of the minimization algorithm of producing a concise representation of a given set of privacy constraints we performed the tests depicted in Figure 4.9a and Figure 4.9b. We generated a random set of constraints on a relation R with $k = 15$ attributes and $n = 100$ tuples and we applied on them the minimization algorithm. We count the number of privacy constraints and of conditions in which the input set is reduced by the minimization. We can see that the minimization is performing quite well in reducing the number and length of the set of privacy constraints. Obviously the algorithm cannot return more privacy constraints with more conditions than the set given in input.

4.6 Conclusions

In this chapter we present an approach to elicit and specify privacy constraints based on sampling. The idea is to propose a view of the database to be protected to a privacy



(a) Minimization of input constraints.

(b) Minimization of input conditions.

Figure 4.9: Constraints Minimization applied on 15 attributes and 100 tuples.

expert, with the goal of collecting a list of privacy violations by tracking which attributes and tuples the user considers as privacy violations. This way the user will tell us what she want to be concealed in reports produced from that database, but without providing any explanation of why this happens.

We propose an approach to derive from the list of violations a concise definition of the privacy constraints in form of SQL select-project query. The key innovation of this approach is that it is based on the idea of showing a view (called *sample*) of the original database data content, on which the user can more easily identify the violations and consequently the sensitive data to be protected. The creation of the view that will be proposed to the user is particularly critical, as user's possibility to define a set of privacy constraints is restricted to which data we will present. We propose an algorithm to select a set of samples to be shown to the user in a certain order in such a way to satisfy the following conflicting goals: on one side we would like to minimize the number of samples and of tuples the user has to analyse in order to define the set of privacy constraints she desires; on the other side we would like to cover the database to the largest extent to avoid loosing privacy constraints.

The results are promising and we plan to further investigate on the proposed algorithm to improve their performances and to tune the heuristics depending on the particular data distributions. In addition, it should be investigated more on the robustness of the approach to changes in the database content, in order to avoid repeating the whole privacy elicitation process when new data arrives. We plan also to perform tests with real users to see if the results obtained with our lab tests are coherent with users behaviour and with real data distributions. This will also help us to better tune the size of the samples to users cognitive capabilities, maybe comparing also different interaction devices.

Chapter 5

Case Studies: CSS and SIS-H projects

This chapter presents some real case studies in which the solutions and theories presented in Chapter 3 are applied to develop EHRs with different characteristics. Section 5.1 presents the challenges and lessons learned from the CSS (Cartella Socio-Sanitaria) research project to develop a Social and Healthcare Record for the Autonomous Province of Trento (Italy). The solution proposed for a privacy-aware EHR has been successfully applied on a group of real scenarios and the implementation realized is currently a product released to the province of Trento for further experiments and for the real adoption in the social and healthcare services. Section 5.1.3 shows the application of an argumentation framework to prove the adherence of the EHR system developed in CSS to the privacy regulations. Finally, Section 5.2 shows an EHR solution for developing countries in which privacy and data quality coexist with stringent organizational and technical requirements. The results presented here allows to prove the solutions proposed in this thesis are applicable and usable in real scenarios and raised new research challenges.

5.1 CSS Project

In the Trentino region, as in the rest of Italy and in many countries throughout the world, health and social services do not share information systems. The welfare agency delivers its services through many smaller agencies and municipalities, which also do not share information systems and processes. As a result, obtaining visibility on the quality and the economy of service delivery requires integrating literally dozens of completely heterogeneous and fairly complex systems owned and managed by different institutions. In this section, we present a real case study based on the CSS project (Cartella Socio Sanitaria) undertaken by the autonomous province of Trento (Italy) with a dozen of

partners from the IT sector, the public administration and the healthcare services. The project aims at partially automate the integration between social and healthcare institutions in the delivery of social and health services to individuals in need.
The first goal of the project is visibility and accountability: the province is interested in reporting information 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. Today, these indicators are collected manually, sporadically, and with different practices at each institution. The result is that a lot of time is spent to compute them and, in addition, the results are unreliable. This means that at the outset, the initial request was for a data integration project.
A second goal, not initially stated but that emerged during the project also as a consequence of the approach we took - as discussed later in detail - is the partial automation of the cross-organizational processes required to perform the services.
While the first goal was of interest to the province only, the second was very instrumental to actively involve the other participating institutions, as they had the possibility of executing their processes in a faster, more reliable, and cheaper way.

From a technical and organizational perspective, this kind of project is very challenging, and in fact it was the first of this kind in Italy to complete successfully. The interesting aspects are that:

i) it has all the “traditional” challenges of data integration projects with the added complexities of being cross-organizational and characterized by a large number of medium and small institutions that also dynamically grow over time (civic centers, hospitals and social care institutions will progressively join the integrated platform now that the validation phase is completed);

ii) such an integration in a traditional data warehousing approach is not viable due to privacy rules which simply forbid to extract data and put it in a central repository so that it can be used for analysis.

Indeed, there are strict legal constraints on the way data is collected, stored, distributed and in general processed in a context with multiple data controllers and accessed by multiple types of data consumers (medical staff, administrative staff and government). Such constraints make it difficult to identify application protocols and policies for such kind of data integration.

The project presented here addresses specifically these problems and in particular aims to:

i) automatically obtain information about Key Performance Indicators (KPI) and cost metrics for the social and health services provided by an initial pilot group of agencies;

ii) be able to easily add other institutions to the initial group after the first pilot.

Point ii) is particularly difficult because the systems used in each organization are very heterogeneous with solutions implemented both in-house by dedicated IT departments (as in municipalities) or acquired by third party IT companies (as in private organizations like elderly houses).

5.1.1 Assistance and Medical Process Analysis

This section presents the analysis approach adopted to understand how assistive processes are managed in the province and Trento and to translate them into IT requirements for an EHR infrastructure: the CSS platform.

The CSS platform derives from an in-depth study of the healthcare and socio-assistive domains conducted in synergy with the local governing bodies represented by the Province of Trento (that will host the system) and the Social Welfare Department (that will use the infrastructure and the business intelligence services), the Health Care Agency, two major municipalities and local companies providing telecare and services for the elders in the Trentino region. In particular, we analyzed, together with the domain experts, some clinical and assistive processes that involve all the partners mentioned above to identify the organizational and technological constraints of the IT systems in use, to capture the business processes executed and the bits of data they produce and exchange with each other. This **process-oriented analysis approach** relieves us from the internal complexity of the single information sources as it focuses only on the 'visible' effects of the business processes to track only the data that the data sources are willing and interested to share.

We created a working group with experts from each institutions and organizations and we spent a considerable amount of time in analyzing the administrative, medical and social domains. This because there is no documentation on the business processes going on among these institutions as, most of the times, they are represented only in people's mind or are left completely fuzzy: the decision about what to do is left to personal judgment. The goal of this analysis phase is to understand how people work and use their IT systems to devise an integration solution which minimizes the impact on their current working practices and to propose improvements to facilitate their work and resolve inefficiencies. There is a lot of work in the social studies to interact with people. Such techniques allow to propose them the more suitable IT device or to automatize certain operations currently performed manually (e.g. ethnography and participatory design approaches like CSCW, Computer Supported Cooperative Work [51]).

In our case study, we performed a series of interviews (*focus groups*) with the domain experts that 'taught' us how to decode their fuzzy and complex world. Our main concern in the analysis was on producing a concise representation of the domain usable both by domain experts to model their world and by designers to get a set of requirements that can be prioritized and rapidly translated into a system specification.

In our analysis we were interested in the *workers-applications* interactions in order to capture the data produced at each step with the twofold goal of: (i) isolating the points of cooperation and interoperability among the inter-related processes executed by the parties and (ii) feeding a Business Intelligence (BI) module.

The BI solution, designed considering such inter-related processes as a whole, enables a comprehensive analysis of the business processes occurring among the parties that in the current context the governing bodies (from now on the Governance) cannot carry out.

We used activity diagrams to model the processes adding to the standard notation some stereotypes to represent the data (both on paper and in the form of informative events) that are produced during each activity (Figure 5.1).

Despite more modeling formalism exists, like for example BPMN (Business Process Modeling Notation [130]), we decided to use a notation as much simple as possible. In particular, BPMN is a powerful graphical modeling language to represent the business processes in a formalism that can be easily translated into executable languages like BPEL [94]. However, in order to do that, the model should be very detailed to clarify any ambiguity. This requires to invest a considerable amount of time in a deep analysis making the approach applicable only on a restricted scenario.

In order to have a complete picture of the domain without the need to analyse so deeply the business processes involved we adopted a more abstract approach. We identified the information of interest for the interoperability and the monitoring on the business processes in form of events.

In doing that we kept also into account the Key Performance Indicators (KPIs) the organizations delivering socio-assistive services use for budget planning and monitoring of the service quality to verify if it is compliant to contractual agreements (e.g. the time elapsed from the approval of Teleassistance service request to its activation cannot be longer than 7 days).

Table 5.1 gives an excerpt of the KPIs for each type of source involved: Province of Trento, Welfare agency, Healthcare agency, Local municipalities and districts, Private companies and no-profit organizations.

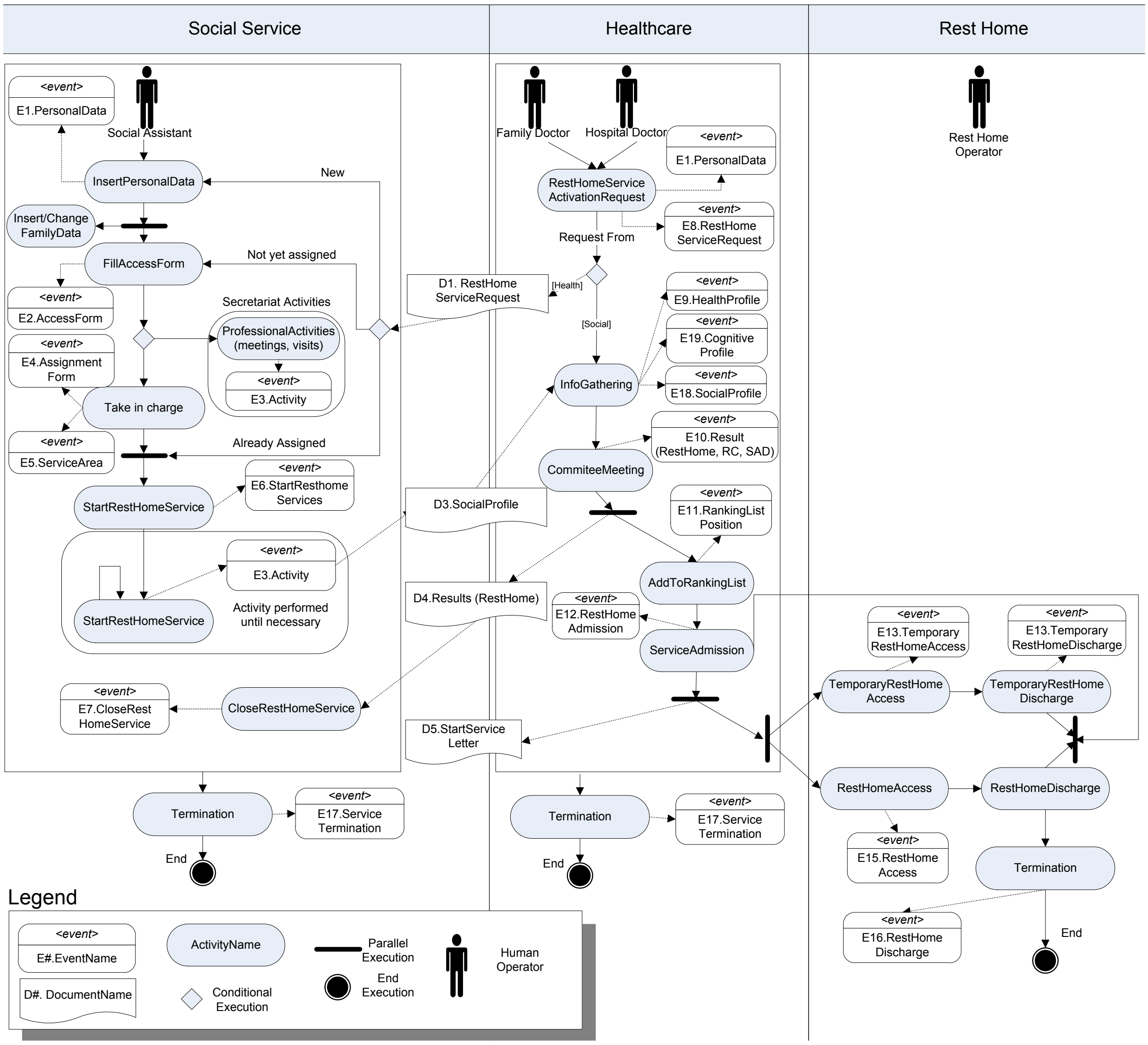


Figure 5.1: Excerpt of *request*, *evaluation* and *provision* process of *Rest Home Service*.

The steps of analysis are summarized below:

1. select reference scenarios and enough data to cover all the involved partners and to produce meaningful BI reports;

2. interview domain experts to understand and model incrementally the business processes to: sketch out the glossary used, actions performed, responsibility (who does what), exceptions to normal flow of actions, constraints and dependencies among different systems to proceed with next steps, input/output data (events) of an action;

3. isolate events that can be produced automatically by the system and their point of origin;
4. isolate the documents exchanged among the parties and assign priorities for their conversion into events;
5. detail the data content of the events: fields, type, optional or compulsory nature, standardized nomenclature and domain attributes;
6. identify the KPI (Key Performance Indicators) to design a Data Warehouse (DWH) and the reports in the BI module. The schema of the DWH is given in Appendix A.

The analysis enabled us to:

- discover, model and document the business processes occurring inside each institution and their inter-relationships despite the poor formalization in healthcare and socio-assistive processes, given that each IT system, and even each single operator, has its own way to proceed inspired by best practices of their reference organization or just by common sense and past experience;
- identify and refine a modeling formalism to summarize the knowledge we obtained from the analysis with the domain experts in concise, tangible and usable documents in an understandable way, even by non-technical people;
- 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 "informatized" knowledge that is more usable;
- isolate events and their contents (fields), domain values, point of generation and frequency of generation; derive KPI to understand what the users (operators of the healthcare and socio-assistive domain and governing bodies) need and their expectations from the system.

The standardization of events among different data producers is what allows their integration. An alternative integration approach is to let the DWH doing the standardization of the event's data at cleaning time (during the ETL -Extract Transform and Load- procedure [97, 98]) but this will limit the use of the events only to the BI activities. Instead, we want to make the events directly consumable by different sources to allow the cooperation among the parties. This requires to the events to speak the same language, that is, to use a common and shared glossary (e.g. a shared nomenclature of social-health services).

Table 5.1: KPIs required by organizations delivering socio-assistive services.

Organizations	Responsibilities	KPI
Province of Trento	Collects information on the services delivered in a provincial data warehouse to monitor their quality, the economic resources employed for reimbursement and budget planning and to monitor the demographic evolution of assistive needs	♯ of assistance requests per district/age classes.
Welfare agency	Evaluates cognitive and social state of the patients and their level of autonomy to complete the requests of activation of socio-assistive services	♯ of patient per social workers; % of accepted requests of assistance; ♯ of requests per territory area; ♯ of services activated within 60 days.
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 practitioner, hospital doctors, social worker)
Local municipalities and districts	Control the administrative process to activate the socio-assistive services with cross-validation of certificates, financial support and delegation of service provisioning to accredited organizations	average cost of services per person; ♯ of administrative practices completed within 60 days
Private companies and no-profit organizations	Deliver the final services (tele-control/tele-assistance, nursing home services, long term assistance in elderly houses or recreation centers) to the patients and interact with their family doctors, relatives and all the network of people connected to the patient	♯ of alarms per type (false/ healthcare/ social alarm, monitoring device failure); hours of nursing home services per patient.

Another reason that encourage us to go for an event-based feeding of the DWH is the complexity of the business processes analysed which makes particularly difficult to create, with a minimal development effort, the ETL procedures capable to extract from the sources such data.

In fact, we saw that the modeled processes have many exceptions: for example the request of access to a rest home 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 of the patient).

To make things even more complex, the same information system is used following a different sequence of activities by multiple municipalities depending on internal procedures and the working practices of the operators. As a consequence some data is not collected at all and the corresponding events cannot be generated because their information is not entered into the system.

From this analysis we saw that, if we consider the different processes from each institution as a whole they result in a highly inefficient composed process. The main weakness is that the same information is duplicated in different systems and re-inserted many times because it is communicated by means of paper documents (e.g. in Figure 5.1 the event $E18 : SocialProfile$ originates from the manual insertion of document $D3$). The analysis work allows to identify some points of improvements:

- the dematerialization of documents by converting the data they carry into electronic events;
- the reorganization of the processes internally and in the way they interact each other to avoid duplicate flows of information;
- the identification of the actors, the data they control, their roles (data producers and data consumers) and the purposes for which they use the data.

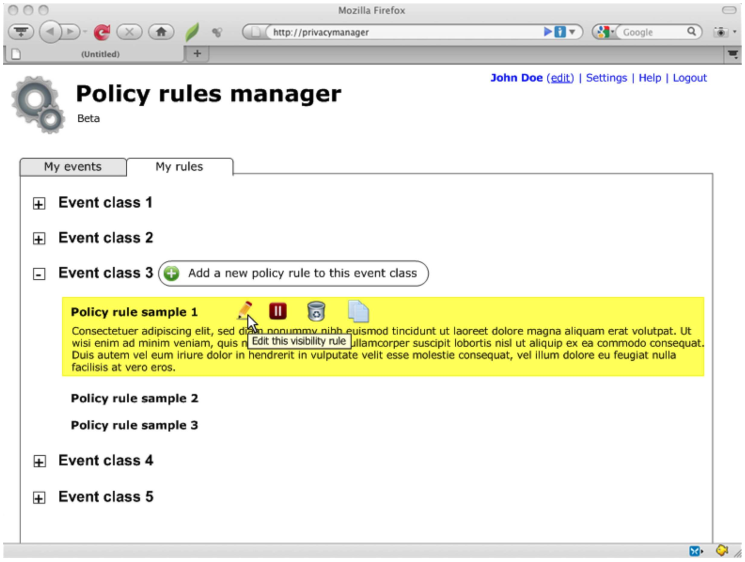
The scenario is particular because it combines data belonging to different domains (social and medical) from multiple data controllers that should cooperate still maintaining their control on the data. This imposes strict constraints on the way data flows among the sources and imposes particular care in managing the data lifecycle and sensitive data by means of a communication protocol which: minimizes the traffic of sensitive information; avoids to store sensitive data in a central place; allows the data owner to decide what can be shown, to whom and for which purposes by means of privacy policies.
The EHR architecture proposed in Section 3.1 is a comprehensive solution for all these requirements and constraints in this scenario, both organizational and technical, and it has been successfully applied in the CSS project[1]. In this section, we are not presenting again the details of the EHR solution but only the approach used to realize it in the real scenario.

5.1.2 The Prototype System

The prototype system of the CSS project has been developed with standard and consolidated open source technologies by putting into practice the architecture in Section 3.1, and is planned to be released as open source code under the LGPL license v3.0[2].

[1] http://www.trentinosociale.it/index.php/Il-nuovo-welfare/E-welfare/Progetto-Cartella-Socio-Sanitaria-CSS

[2] http://www.gnu.org/licenses/lgpl-3.0.html



John Doe (edit) | Settings | Help | Logout

Policy rules manager

Beta

My events | My rules

⊞ Event class 1

⊞ Event class 2

⊟ Event class 3 ⊕ Add a new policy rule to this event class

Policy rule sample 1

Consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis n ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros.

Edit this visibility rule

Policy rule sample 2

Policy rule sample 3

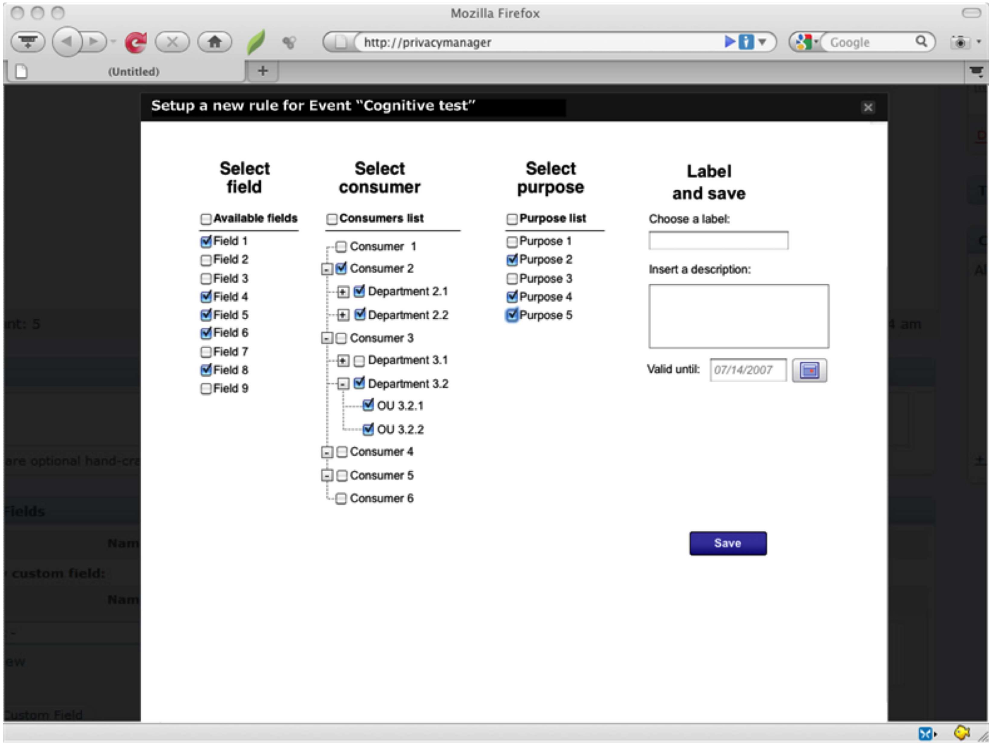
⊞ Event class 4

⊞ Event class 5

Figure 5.2: Dashboard of the Privacy Rules Manager.

The system is basically “transparent” to the end users when it deals with the storage of events and their routing among the different partners, as all these functions are performed automatically by the modules of the underlying cooperation infrastructure and by the legacy systems. The only two features exposed to the end users by means of a front-end are represented by the privacy policy definition and the business intelligence analysis. The business intelligence analysis is performed by a reporting console that is still under development as the underlying data warehouse structure is not yet finalized (see a first proposal in Appendix A). We are particularly interested in the privacy policy definition GUI because it is a very critical point of interaction with the user, and it is where the techniques presented in Chapter 4 can be applied. The GUI developed in the prototype does not use yet the approach presented in Chapter 4 to define the privacy policies through data sampling. However, we plan as future work to introduce also the sampling approach for a test with real users and a comparison with the standard privacy definition approach.

Figure 5.2 shows the Dashboard of the Privacy Rules Manager the data controller (owner of the data) will use to define the privacy policies. The user can define one or more privacy policy rules for each type of event. Figure 5.3 shows the GUI for the definition of an instance of privacy policy. The user can select: i) one or more items from the list of fields in the event details type, ii) whom (i.e. one or more Organizational Unit inside a department as consumers) and iii) the admissible purposes. Privacy rules are saved with a name and a description.



Setup a new rule for Event "Cognitive test"

Select field

□ Available fields
☑ Field 1
□ Field 2
□ Field 3
☑ Field 4
☑ Field 5
☑ Field 6
□ Field 7
☑ Field 8
□ Field 9

Select consumer

□ Consumers list
- □ Consumer 1
- ☑ Consumer 2
  - ☑ Department 2.1
  - ☑ Department 2.2
- □ Consumer 3
  - □ Department 3.1
  - ☑ Department 3.2
    - ☑ OU 3.2.1
    - ☑ OU 3.2.2
- □ Consumer 4
- □ Consumer 5
- □ Consumer 6

Select purpose

□ Purpose list
□ Purpose 1
☑ Purpose 2
□ Purpose 3
☑ Purpose 4
☑ Purpose 5

Label and save

Choose a label:

Insert a description:

Valid until: 07/14/2007

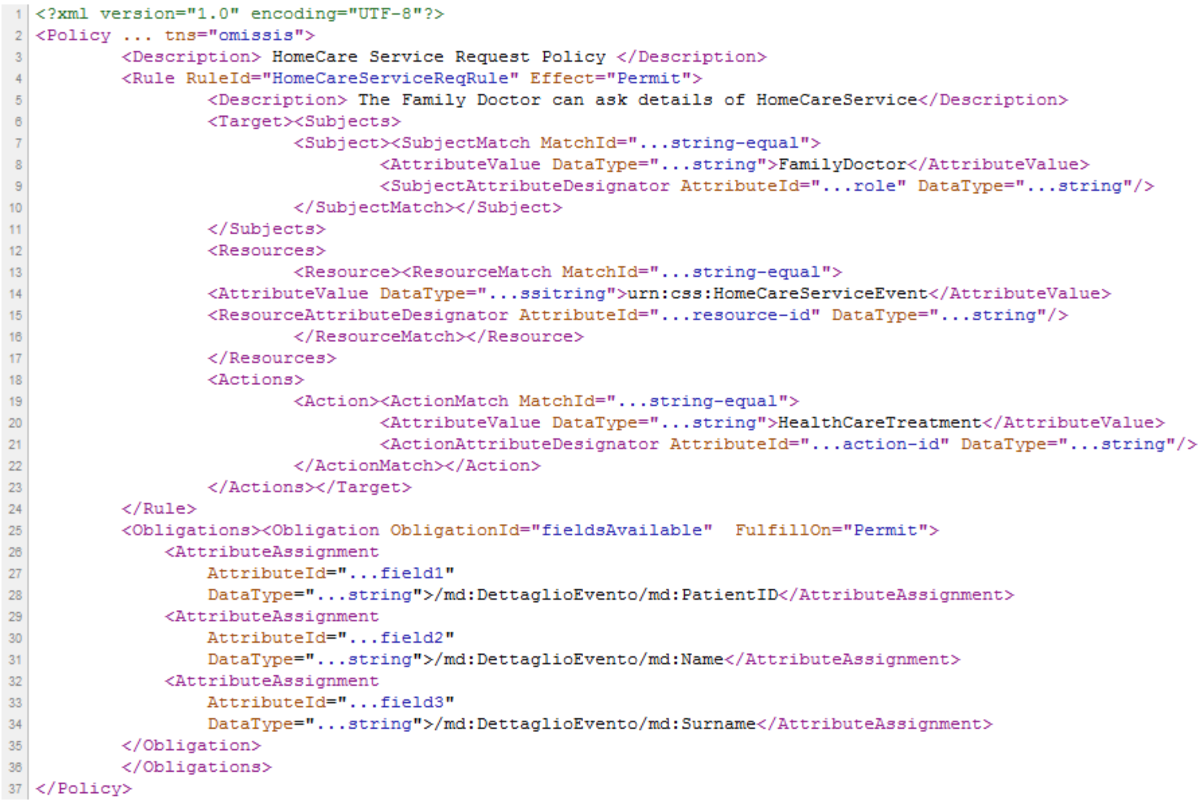
Figure 5.3: Privacy Policy definition tool.

Optionally the user can specify a validity date that limits the application of the rules to a certain time window. This option is particularly useful when private companies are involved in the care process and should access to the events of their customers only for the duration of their contract.
Some of the advantages of the GUI are listed below:

- it is very intuitive to use as it does not require any knowledge of XACML but it asks to the user to define a policy in terms of actor, type of event to protect and admissible purposes of use;

- it automatically generates and store in a policy repository the privacy policy in XACML format.

In Figure 5.4 we provide an example of a privacy policy that allows a user with role family doctor (lines 7–10) to access the event of type *HomeCareServiceEvent* (lines 13–16) for *HealthCareTreatment* purpose (line 20). In particular, only the fields *PatientId*, *Name* and *Surname* of the details are accessible (line 25-36).



```
<?xml version="1.0" encoding="UTF-8"?>
<Policy ... tns="omissis">
        <Description> HomeCare Service Request Policy </Description>
        <Rule RuleId="HomeCareServiceReqRule" Effect="Permit">
                <Description> The Family Doctor can ask details of HomeCareService</Description>
                <Target><Subjects>
                        <Subject><SubjectMatch MatchId="...string-equal">
                                <AttributeValue DataType="...string">FamilyDoctor</AttributeValue>
                                <SubjectAttributeDesignator AttributeId="...role" DataType="...string"/>
                        </SubjectMatch></Subject>
                </Subjects>
                <Resources>
                        <Resource><ResourceMatch MatchId="...string-equal">
                <AttributeValue DataType="...ssitring">urn:css:HomeCareServiceEvent</AttributeValue>
                <ResourceAttributeDesignator AttributeId="...resource-id" DataType="...string"/>
                        </ResourceMatch></Resource>
                </Resources>
                <Actions>
                        <Action><ActionMatch MatchId="...string-equal">
                                <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">/md:DettaglioEvento/md:PatientID</AttributeAssignment>
            <AttributeAssignment
                AttributeId="...field2"
                DataType="...string">/md:DettaglioEvento/md:Name</AttributeAssignment>
            <AttributeAssignment
                AttributeId="...field3"
                DataType="...string">/md:DettaglioEvento/md:Surname</AttributeAssignment>
        </Obligation>
        </Obligations>
</Policy>
```

Figure 5.4: Example of Privacy Policy.

An on-field three phase experimentation has been performed with:

1. Deployment: the definition of a deployment plan with the partners (3 months)
2. Testing and Evaluation: the deployment plan actuation and system evaluation in a controlled environment (6 months)
3. On field experimentation: the evolution from prototype to product at the provincial data center to operate in the real environment (1 year).

The system has completed the evolution from prototype to product and is approaching the on field experimentation phase. The deployment plan has been defined with the Province and the other partners and refined on the base of a first round of tests to verify if the system is properly dimensioned wrt the number, size and rate of production of the events (some figures are shown in Table 5.2) and IT infrastructures available at the sources. The goal is to limit the effort of the partners to interact with the interoperability infrastructure. Figure 5.5 shows a sample legacy systems used by the parties in the experimentation.

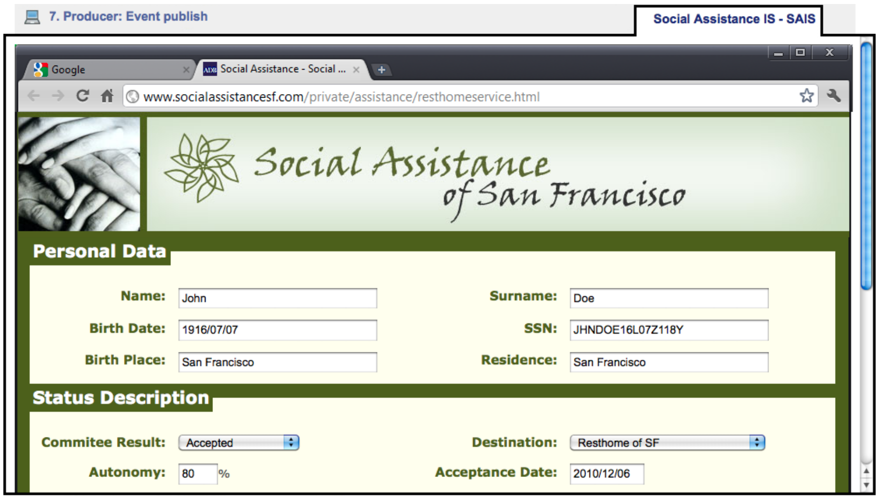
The two core outcomes we expect to evaluate from the adoption of this system are:

- to achieve parties' interoperability introducing new capabilities in their systems in terms of data becoming sharable among the parties;
- to perform business intelligence on complex inter-company business processes.

In the first evaluation phase we defined some fictitious citizen profiles to execute a set of complete request-evaluation-provisioning process executions in the real systems and verify the correct production, routing and consumption of the events to compute the KPI of the DWH. Thought the KPIs obtained by 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. In the on-field experiments the real data of patients and citizens will be used.
The socio-medical services selected for the experimentation are listed below:

- Assistive and healthcare services for elderly and disabled people provided directly at home with nurses, family doctors, social workers and employees of private cooperatives for meals delivery and house cleaning services;
- Long term healthcare services in specialized structures like Rest Home;
- Recreation centre for elderly people providing transportation, daily activities and meals supply;
- Tele-control and Tele-assistance services with a 24h call center checking periodically the state of the assisted person and in case of critical problems (or of direct request of help from the user) activates emergency services (ambulance) or notifies the abnormal behavior to reference people (e.g. relatives, neighbors, general practitioner and social workers).

These services are characterize by a mix of social and medical partners contributing together to the delivery of the services. In that way it is possible to verify and prove the capabilities of the infrastructure to allow entities from different domains to cooperate.
The effort required to join the cooperation infrastructure and to share information from the technological point of view was very low since partners 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 business logic of events storing, requests resolution, privacy policy enforcement, listening for incoming messages and the security protocols are provided by the Local Cooperation Gateway that we release to the partners.


7. Producer: Event publish
Social Assistance IS - SAIS
Google
Social Assistance - Social ...
www.socialassistancesf.com/private/assistance/resthomeservice.html
Social Assistance of San Francisco

Personal Data

Name: John
Surname: Doe
Birth Date: 1916/07/07
SSN: JHNDOE16L07Z118Y
Birth Place: San Francisco
Residence: San Francisco

Status Description

Commitee Result: Accepted
Destination: Resthome of SF
Autonomy: 80 %
Acceptance Date: 2010/12/06

Figure 5.5: Example user interface for information system in an elderly house.

Table 5.2: Estimation of the number of events and data exchanged yearly.

Partner	Number of events	Estimated number of instances	Total event dimension(MB)
Healthcare Agency	9	14500	56
Welfare Agency	14	74700	251
Local Municipality	7	53100	205
tele-control/ tele-assistance	7	93985	322

This was one of the key factors for the EHR success as it minimized considerably the partners' effort in a scenario in which high learning curve and entry barriers are a deterrent for small private institutions.

The biggest amount of time was required to engineer the events and check the availability of detail message's data in the DBs at the legacy systems.

We defined about 40 events to cover the same number of documents with no need to deal with the DB at the sources that normally have more than 100 tables, some having more than 50 attributes. This greatly simplified the work and allows to accelerate the development. The structure of an event is defined in an XSD that is uploaded on the central infrastructure with an intuitive wizard. There is no need for any further configuration apart from the creation of the XSD schema. The discovery and subscription of events is very simple and intuitive and it takes just 10 minutes with the support of the GUI as shown in the demo at [19].

During the testing and evaluation phase we identified and faced practical problems like the definition of the privacy policies and their formalization into contracts acceptable also by the Italian privacy Guarantor. The GUI that simplifies the policy definition step enables partners to define very fine-grained exchange rules over their data. In this way they have 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 partners and in this case it will be necessary more than one policy. All policies are defined following the national regulations in sharing health data [88] and helps to demonstrate to the privacy Guarantor that the project is following correctly the privacy regulations [57]. In the next section is shown a modeling formalism that can be applied to assess the compliance of the EHR architecture to these privacy regulations.

5.1.3 Privacy Requirements Compliance

An EHR is a very critical system by the nature of the data and processes it manages and the risks of incidents in the medication process is well known [93]. For this reason its use should be conditioned to a set of procedures and best practices devoted to reduce and manage the risks of medical errors [92, 107]. Also the privacy of patients should be preserved from the risks of privacy breaches. For these reasons, before adopting an EHR solution, the organizations participating in the healthcare process carefully evaluate the system and ask for certifications released from the administrative bodies.
In this work, we focused on the privacy regulations and on the authorizations released by the privacy guarantor office required to realize and then to adopt an EHR solution. In order to obtain such certifications the software designers should describe to experts in the medical domain and in the law, but typically not in IT, how the system is designed and to prove it is compliant to the privacy regulations.

In this section, we show the results of a joint work with FBK presented in [22] to apply an argumentation framework together with a goal oriented requirement engineering technique on the design of an EHR and specifically on the application of the architecture presented in Section 3.1 in the CSS project.
A graphical formalism and a requirement engineering methodology are used to describe and motivate why certain design choices have been introduced and to prove, by means of argumentation chains, that the EHR is compliant to requirements derived from the privacy law.

As seen in Section 5.1.1 the medical and socio-assistive processes involve many actors from the public and private sectors. The interactions among these parties are regulated by laws governing how information flows inside the public and private companies and also among them [25, 26, 6]. Public bodies, like the municipalities, developed some guidelines (in [121, 120]) that further specify the general provisions at national level (see [88, 57]) and provides some operative indications for their employees.

Table 5.3: Laws from Guidelines on EHR [88] and Personal Data Protection Code [57].

Name	**Description**
L1	criteria should be laid down to encrypt and/or keep separate the data suitable for disclosing health and sex life from any other personal data; [..] As for EHRs, secure communication protocols should be deployed by implementing encryption standards for electronic data communications between the various data controllers.
L3	The Electronic Health Record should be set up by prioritizing solutions that do not entail duplication of the medical information created by the health care professionals/bodies that have treated the given data subject.
Dlgs 196/2003 n.26(1)	Sensitive data may only be processed with the data subject's written consent and the Garante's prior authorisation, by complying with the prerequisites and limitations set out in this Code as well as in laws and regulations.
Dlgs 196/2003 n.26(2)	The Garante shall communicate its decision concerning the request for authorisation within forty- five days; failing a communication at the expiry of said term, the request shall be regarded as dismissed. Along with the authorisation or thereafter, based also on verification, the Garante may provide for measures and precautions in order to safeguard the data subject, which the data controller shall be bound to apply.

An excerpt of the more important laws concerning an EHR from the Guidelines on EHR [88] and the Personal Data Protection Code [57] is listed in Table 5.3. The EHR infrastructure was designed with these regulations in mind but many decisions are based on implicit assumptions based on the knowledge of the domain and of the scenario and are not clearly documented in the project documentation. This makes difficult to explain the solution to the administrative bodies to convince them about the compliance of the system to the regulations. We need a way to isolate the architectural elements of the solution that impact on the satisfaction of the regulations and to map them to the laws they derive from.

In the approach proposed in [22] we start from the goal model of the system focusing on the goals that are more controversial from the privacy point of view (e.g. the access and retrieval of patient's data). We use a law-driven framework named *Nòmos* [141], an extension of the *i** [160] goal-oriented modeling language, that adds the capability to model legal concepts to explicitly state which laws the goals derive from.
A *Nòmos* model for the EHR system on the more critical privacy laws of Table 5.3 is reported in Figure 5.6. In particular, law $L1$ asks to *prioritize solutions which do not require to duplicate any sensitive information outside the boundaries of the data controller.* Law $L3$ imposes the *separation between administrative and medical data.*

Table 5.4: Argumentations.

Name	Description
A1.1	EHR does not know what is sensitive and what is public
A2.1	EHR is not said to be a trusted party
A3.1	Data Controller role undefined
A3.2	Delegate duplication not cited as solution
A3.3	Duplication admitted if no other choices available
A3.1.1	Healthcare professional/bodies are data controller by law 26(1)(2) of Dlgs196/2003
A3.2.1	No duplication outside data controller boundaries
A3.2.1.1	Duplication allowed but agreement needed
A3.3.1	Evaluated different solutions

The creation of a *Nòmos* model allows to make explicit certain decisions taken at design time to comply with the privacy regulations that so far was just in the mind of the designers, for example: goal $G1$ states the fact that different authorizations profiles can be achieved with privacy policies and goal $G2$ indicates that to avoid duplications the sensitive data is maintained at the data controller. Usually, analysts interact with the users to validate and evolve the model until a common agreement is reached among all.
When the system is complex and when there are many actors contributing to the project, like in the case of an EHR, it is difficult to get an agreement on such decisions from all of them and, most importantly, from the privacy guarantor office.
The approach presented in [22] applies an argumentation framework [95] to interact with the users in a more systematic way.

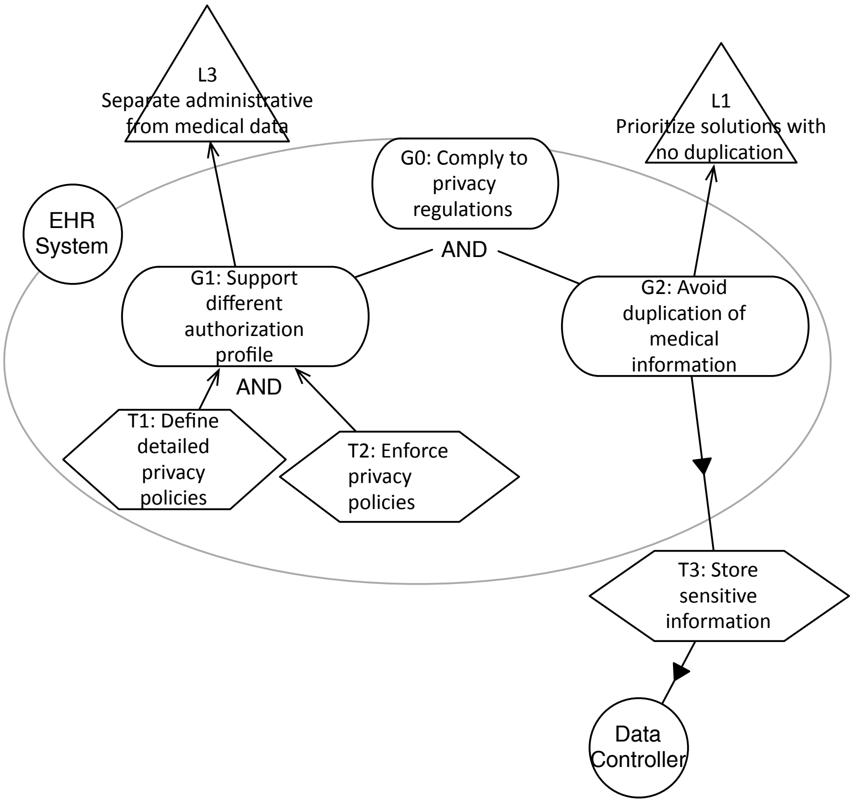

L3
Separate administrative from medical data
L1
Prioritize solutions with no duplication
G0: Comply to privacy regulations
EHR System
AND
G1: Support different authorization profile
G2: Avoid duplication of medical information
AND
T1: Define detailed privacy policies
T2: Enforce privacy policies
T3: Store sensitive information
Data Controller

Figure 5.6: An excerpt of the compliance for the EHR actor, expressed by means of a *Nòmos* model.

In particular, it uses the *Nòmos* notation to link requirements with the legal concepts they derive from and combines this with the ability to trace the reasoning process performed together with the actors to reach an agreement on the model by means of argumentations. As defined in [95] an "*argument is a piece of information (e.g., a statement) that either provides support for, or is provided against choosing an alternative, where an alternative is a potential solution to the stated problem*". In practice, the feedback of the user is captured by means of argumentations that can defeat or support a model entity (e.g. a task, a goal or another argument).

An example of such an argumentation process is shown in Figure 5.7. The argumentations are represented as rectangular notes connected to model entities or to the argumentations they try to attack.

As the argumentation process goes on by repeated interactions with the user a chain of argumentations is created.

The process ends when no more arguments in favor or against a model entity are added.

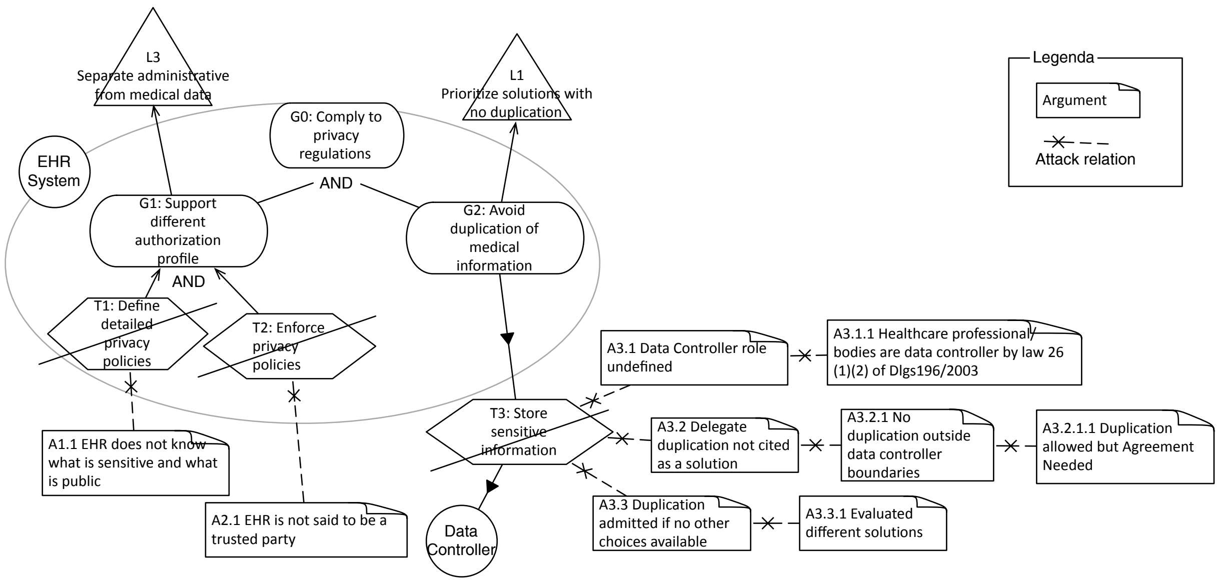

L3
Separate administrative from medical data
L1
Prioritize solutions with no duplication
Legenda
Argument
Attack relation
EHR System
G0: Comply to privacy regulations
AND
G1: Support different authorization profile
G2: Avoid duplication of medical information
AND
T1: Define detailed privacy policies
T2: Enforce privacy policies
T3: Store sensitive information
A3.1 Data Controller role undefined
A3.1.1 Healthcare professional bodies are data controller by law 26 (1)(2) of Dlgs196/2003
A3.2 Delegate duplication not cited as a solution
A3.2.1 No duplication outside data controller boundaries
A3.2.1.1 Duplication allowed but Agreement Needed
A3.3 Duplication admitted if no other choices available
A3.3.1 Evaluated different solutions
A1.1 EHR does not know what is sensitive and what is public
A2.1 EHR is not said to be a trusted party
Data Controller

Figure 5.7: Attacks modeling. The existing *Nòmos* model and the set of arguments attacking model entities and other arguments.

The argumentation approach is divided into three steps: 1. argumentation phase, during which the argumentations are collected from the users; 2. justification phase, in which the argumentations chains are analysed to see which model elements are defeated; 3. model evolution phase, in which the objections to the model surviving to the justification phase are resolved by changing the model.

The argumentation phase will continue until no other arguments are added. In the case study of the EHR system, we performed three rounds of discussions with the analysts and designers of the system and the sequence of argumentations produced is reported in Table 5.4. In the following we describe how we dealt with these argumentations and how the model evolves.

The first two arguments ($A1.1$ and $A2.1$) defeat the goal $G1$ as the solutions proposed are based on wrong assumptions and in particular: $T1$ assumes the data processor (that is the EHR system) knows which privacy policies to enforce but this is not true as the EHR system deals only with the transmission and indexing of the events without considering the semantics of the data (argument $A1.1$); $T2$ assumes the EHR system is a trusted party which can enforce the privacy policies on behalf of the data controller but this is not the case if the data processor is not under the control of a certified authority like the Province but is a private company that is hosting the service (argument $A2.1$). These two arguments by defeating the goals highlight that the system is not compliant to the privacy regulation.

To resolve these objections, the model is modified by delegating the defeated tasks to the data controller because it has the ability to define the privacy policies and is also entitled to enforce them. This model corresponds to the *decentralized enforcement* configuration described in Section 3.2. The argumentation $A3.1$ attached on task $T3$ is related to the role of the data controller that was not explicitly stated. This objection was neutralized by the argument $A3.1.1$ citing the law 26(1)(2) of Dlgs196/2003 which explicitly defines the role of healthcare bodies and their duties as data controllers when collecting and processing sensitive data. According to such laws the data controller is authorized to use sensitive data with the "*data subjects written consent and the Garantes prior authorization*". The Healthcare agencies collect the consent to use the sensitive data from the data subject and they also provide a declaration to the Garante to get the authorization. From this we derive that the healthcare agencies are responsible of the use of the data and they assume the role of data controller. This resolves the argumentation chain without introducing any change in the model.

The other objections to the task $T3$ originate more complex argumentation chains and requires three interactions with the users to reach an agreement. Task $T3$ assumes that to avoid the duplication of medical information is enough to delegate the storage of such information to the data controller. However, as stated by argumentation $A3.3$ this solution does not give evidence of the fact that different solutions have been evaluated as required by law $L1$ neither that storing the data at the data controller is an admissible solution by the law ($A3.2$).

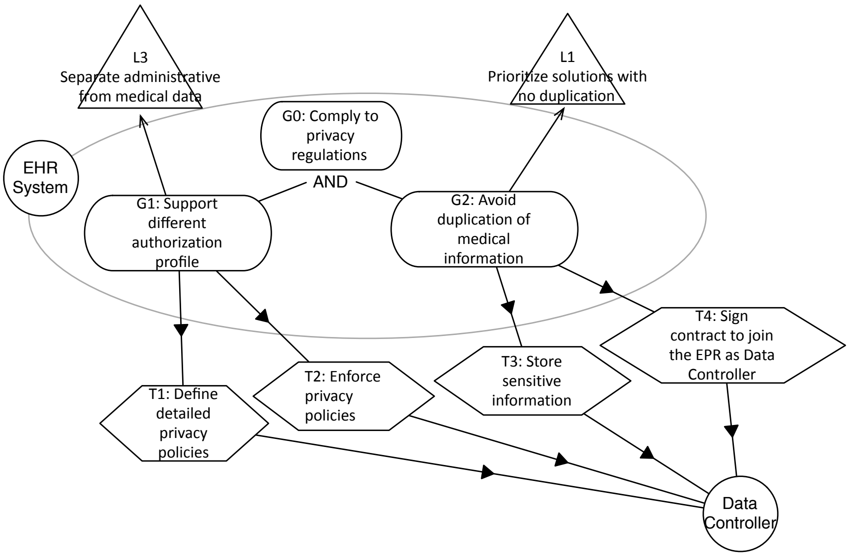

L3
Separate administrative from medical data
L1
Prioritize solutions with no duplication
G0: Comply to privacy regulations
EHR System
AND
G1: Support different authorization profile
G2: Avoid duplication of medical information
T4: Sign contract to join the EPR as Data Controller
T3: Store sensitive information
T2: Enforce privacy policies
T1: Define detailed privacy policies
Data Controller

Figure 5.8: Extended model. The evolution of the model on the bases of the arguments related to the existing *Nòmos* model.

In the second round of discussion, argument $A3.3$ was neutralized with an explicit statement indicating that different solutions have been evaluated before the final design of the system and documented in the project documentation (argument $A3.3.1$ "Evaluated different solutions"). This resolves the argumentation chain with no impact on the model.
Argumentation $A3.2$ requires to make explicit the assumption of the designers that law $L1$ does not allow duplication outside the boundaries of the data controller and consequently to be compliant is enough to leave the storage of sensitive information to the data controller itself. However, this argumentation is not enough to convince the user that the goal is reached and another argument ($A3.2.1.1$) is attached saying that the solution proposed is accepted if an agreement is signed with the data controller. The line of arguments $A3.2$, $A3.2.1$, $A3.2.1.1$ produces a valid attack to the model (in particular to the task $T3$) and requires to extend the model as shown in Figure 5.8. The task $T4$ "Sign contract to join the EHR as Data Controller" neutralizes the last argumentation and makes explicit that the data controller agrees to retain the responsibility in storing the sensitive data and in enforcing the privacy policies to make sure only authorize data will be delivered to the consumers.

5.1.4 Lessons Learned

In this section we discuss how to develop an interoperability solution for the the cooperation of organizations in the social and healthcare domain in the real scenario of the province of Trento. The interoperability infrastructure presented in this case study is now being put into production after a successful validation and experimentation phase with the test partners and institutions [18]. The solution allows us to couple the benefits of a pub/sub event-based system (decoupling of publishers and subscribers) with a privacy approach that is compliant with the privacy laws typically adopted in managing healthcare information, as validated with privacy compliance experts. The system provides services for policy definition and application but does not dig into the data that travels from sources to destinations and improves the control on data exchanges and consumption by validating and logging all data requests and exchanges.
The informatization of data flows permits also to avoid privacy violations caused by manual importing into systems (operators does not input anymore data into systems). The privacy requirements elicitation tool evaluated in this case study is only a fist prototype that will be further improved to allow privacy experts to define privacy policies in a very intuitive way without knowing the underlying DB schema. In the future, we plan to make this GUI even more user friendly by exploiting the privacy policy elicitation and specification approach described in Chapter 4.

In the second part of this case study we show how to trace by means of argumentations the mental processes of the designers and of the users to reach a certain requirement. This avoids situations in which during the analysis the same requirements are re-discussed many times because is lost the reasoning and the motivation of their introduction. By running through the model enriched with the argumentations together with the users, it is easier to reach an agreement on a stable model and to prove the system is compliant to the regulations. In this work, only privacy regulations are considered but the approach can be applied to any kind of laws and best practices (e.g. risk best practices [92]).
Indeed, the approach requires more effort to the analyst especially on complex system in which the goal model is already complex and may become unmanageable if it is necessary to trace the argumentations. The complexity of the argumentation framework depends on the number, lenght and 'persistence' of a chain before it is resolved with new argumentations or with a restructuring of the model.
In conclusion, the argumentation framework is useful when applied on well delimited scenarios and on the more critical aspects for which such additional effort is justify by the need of a formal proof or motivation of the compliance of the design choices with the regulations (like the case study analysed in this section). The outcome of the analysis has been used to prepare the documentation required by the privacy guarantor office and particularly to motivate certain design choices with the link to the regulations.
The argumentation framework was applied at the end of the analysis and design phase to validate the solution. However, the approach can be much more effective if it is applied from the beginning of the analysis and during the design instead of just at the end of these phases. In this way, it is possible to change the model of the system during its definition and to understand early core requirements that will have a big impact on the design and implementation of the system.

5.2 SIS-H: Adapt Italian EHR to Mozambique context

In [20] we developed for the *SIS-H* project a generic communication infrastructure for Mozambique hospitals to capture, communicate and analyse clinical events. The goal was to collect statistical information on the distribution of diseases in developing countries and in particular in Mozambique. *SIS-H* allows nurses and volunteers in remote points of care to enter the data on the patients arriving and leaving the hospital like the date and time of arrival and the diagnosis according to the *ICD-10* standard [157]. The system sends the data collected to a *SIS-H* installation located at points of care at higher level in the healthcare organization like central hospitals or the government body. It allows to export the data and save them on a memory stick so they can be transferred to the destination system even without internet connection.

A further extension of the system will manage child vaccinations by means of mobile phones. The idea is to collect the data about newborns from the medical staff registering the event. This data is used to populate a central registrar of children and to plan the vaccination schedule. Reminders are sent to the family of the child and to reference people (e.g. nurse at the point of care, head of the village). When the vaccination is given the nurse sends (always via the mobile phone) the data on the vaccination just given and observations (e.g. bad reactions, reasons why the vaccination was not given).

Designing an information integration solution for healthcare in developing countries is particularly complex due both to technical and organizational constraints. Classical data integration solutions based on a central database or data warehouses are not easily adaptable to this sparse environment that has a high number of data sources acting basically autonomously with no central controller responsible for the creation and maintenance of a central repository.

In such a distributed and loosely connected environment (due to the lack of internet connectivity), privacy and data quality becomes particularly challenging as it is neither possible to control how data is used nor to measure its level of quality. Errors in using data and the occurrence of data anomalies are difficult to detect and resolve unless the effects of the errors become tangible. This may produce a loss of reputation, money or even worst, human lives.

The system in [20] has been developed for the Mozambican Ministry of Health (MISAU) and allows to share healthcare data in a distributed and poorly connected environment to better coordinate healthcare services, minimize human errors, accelerate operative procedures and improve visibility of distributed healthcare processes to the governing bodies. Indeed, it is not already an EHR but it is a first step toward it. This section presents an extension presented in [45] of the basic architectures developed in Mozambique [20] and in Italy for the creation of a distributed EHR with privacy and data quality support. The solution is still under development and this section presents the idea underpinning the research work.

5.2.1 The Challenges

Healthcare services are often composed of critical activities that need to: (i) comply to governmental rules and, (ii) react in a timely manner to changes in citizens' needs. Typically there is a lack of visibility of the services delivered (e.g. how many vaccines are delivered to which categories of citizens) and a gap between the service providers and the service consumers (citizens cannot provide an evaluation to the governance on the service received and vice-versa). This requires a seamless way to collect and share

socio-sanitary information to help both, caregivers (e.g. general practitioners, nurses and volunteers) and governmental parties, to coordinate and simplify their work. This is quite a challenging problem as often different countries, organizations and companies are involved and it is difficult to find an agreed and simple solution that works for everyone. In addition to the classical information integration issues due to the heterogeneity of the data sources, patients are often worried in sharing their data especially if related to their health: stigmatization for diseases such as HIV is a common concern. Secondly, a mistake in the way healthcare data is processed results in economical losses (e.g. incorrect financial report) and, even worse, it may compromise life of people (e.g. use of the wrong therapy). From our experience with similar problems in Italy [18] and in a developing country like Mozambique [20], we understood that solutions based on a central database or data warehouses are not feasible in highly heterogeneous environments. Instead, we propose a more flexible and scalable distributed infrastructure with many nodes managing the data that synchronize each other only with limited information (events) exchanged when needed and if the connectivity is available.
The proposed architecture is an extension of the *SIS-H* system presented in [20] and is still under development [45]. A not exhaustive list of challenges that have to be faced in developing an EHR in developing countries is listed below:

- *Organizational and technical context*: there is no stable and pervasive ICT (Information and Communications Technology) infrastructure or a unique centralized controller as in the solution presented in Section 3.1. The territory is wide and highly heterogeneous. Besides, it is difficult to find well defined operative procedures and/or information systems at the sources. The technical requirements and organizational changes should be minimized as also the data traffic between the nodes of the infrastructure that should act as independently as possible.

- *Data quality issues*: data is collected mainly manually on documents with no unique way to identify individuals as an official patient registry is missing. This compromises quality and credibility of data and requires data analysis (e.g. patient disambiguation techniques) and quality monitoring to measure the quality of exchanged data.

- *Privacy issues*: the use of sensitive information requires to collect from the data subject the consent in sharing and processing it, with a fine grained control on the way it is accessed (by role and purpose) and disseminated. Auditing should be supported as well, by tracing access requests.

- *Support different caregivers*: healthcare services are provided by many actors (like general practitioners, nurses, volunteers) with different skills and education levels.

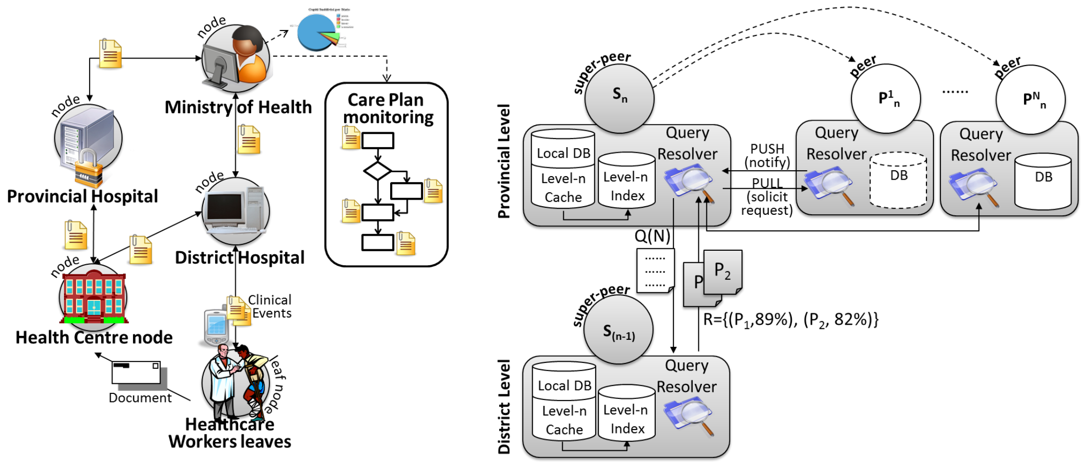
> In general, data is collected manually and it is likely to have mistakes as people are poorly trained and often poorly motivated to follow certain procedures just to comply to policies imposed from the higher levels of the organizational structure, especially if these are time consuming and cumbersome. The devised solution should be simple, intuitive and natural to use so that people are motivated to insert realistic and correct data and to provide feedback. The solution should opt (where possible) to interaction systems collecting data automatically from the environment to minimize the possibility to introduce errors (like the geographical position of the person with a mobile phone).

In order to address the challenges above, we propose an open source interoperability solution for the creation of a distributed EHR, based on the notion of clinical events with a tight control on the quality and privacy of the disclosed information. The architecture is modular, with functionalities exposed as plug-ins that can be easily configured to adapt to the environment and the capability of the deploying node.

5.2.2 EHR for Developing Countries

As for the EHR architecture in Section 3.1 the EHR for developing countries is also based on EDA with nodes, corresponding to healthcare units at different level in the organization, that communicate by means of events as shown in Figure 5.9a. As defined in Section 3.1, an event signals that something important happened to an individual (e.g. the hospitalisation in an healthcare structure) that could be of interest to a caregiver even from a different organization. Nodes are typically located at different levels of the healthcare organization hierarchy. In the scenario depicted in Figure 5.9a the root node is at the Ministry of health and the lower levels nodes are located in: Provincial hospitals, District hospitals and Health Centres. The caregivers (e.g. doctors, nurses, volunteers) are at the leaf nodes and they typically provide data to the higher levels of the hierarchy on documents or using more advanced mobile terminals.
In this scenario, there are no privacy regulations forbidding to store sensitive data in a central place and so there is no need to differentiate among notifications and details as in the EHR for the CSS project [18]. Detail events are maintained in the system (including potentially sensitive information) and fine-grained privacy policies control how sensitive data is released with the same access control approach used in CSS.

As Figure 5.9b shows each level of this interoperability architecture has a “special” node that knows all the other peers: the *super-peer*. A super-peer node should give suitable guarantees of availability and of quality/reputation of the data maintained, computational power and HW resources.



(a) Organizational structure.

(b) Peer configuration and query resolution strategy.

Figure 5.9: Interoperability architecture based on EDA for healthcare plan monitoring.

The EHR module deployed at the super-peer is in charge of:

- managing publish/subscribe events to receive and in case distribute events coming from other peers at the same level and maintaining an index of the events with reference to the source peer and an indication of the data freshness;
- resolving data requests from peers at the same level or from super-peers at other levels (the query resolver engine);
- populating a cache with event requested and their freshness indicator;
- collecting and applying filtering rule (privacy policies) to restrict data access only to authorized nodes based on the roles and purposes of use.

A peer node is equipped with an EHR module configured to provide the functionalities listed below:

- maintains a repository of clinical events;
- provides manual data entry, import/export of events, EHR exploration and quality control;
- applies business intelligence tasks on a data warehouse loaded with clinical events.

The functionalities above are provided also by the super-peer that in this way can compensate gaps in other peer node when needed. For example, the repository of clinical events at the nodes is an "optional" element that in some poorly equipped nodes can be missing. In that case, the super peer can act as a repository for the peer (EHR outsourcing).

Node synchronization is achieved by means of a publish/subscribe mechanism in which a node subscribes to the super-peer to receive events published by a certain data producer. The data producer can accept (or reject) the subscription request and specify privacy policies to filter the events from the sensitive information the subscriber is not authorized to see. In fact, a node is authorized to receive only certain events and only part of them depending on their role and purposes (e.g. governing bodies like the ministry of health may need only aggregated information on the citizens like how many deaths or newborns but not the personal information on the single individual).
A node can query its super-peer to get a certain event on a patient. As it is not said there is a unique identifier of the patient, the query resolver may return a list of events matching some parameters (like name, surname, place and date of birth). The results are provided together with a measure of the matching likelihood so that the user can do an informed choice.
In an ideal world with internet connection always available, events will be delivered to the event subscribers in real-time mode. In our scenario, as we cannot rely on stable connectivity, notifications can be delivered on demand in a pull mode instead of a push mode or also sent in batch as soon as the connection becomes available.
The next section gives a more detailed explanation of the modules deployed in each node, how they are supposed to communicate and the theoretical framework used.

EHR Modules

Figure 5.10 shows in details the internals of the EHR module deployed at the peer nodes. The module is divided into pluggable components that can be configured differently depending on the specific deploying node (super-peers vs "slim" peer).
The main parts of the EHR module are: the *Persistence Manager* with the EHR repository containing the clinical events produced at the node or received from other nodes; the *Routing and Privacy Engine* in charge of delivering events to subscribers and of answering requests for events by a data consumer in compliance to the privacy policies; the *Functional plugins* for publish/subscribe of the events, manual data entry, import/export of events, query resolution and BI analysis. The functional plugins rely on general-purpose layers devoted to privacy and data quality management over events. Specifically, the *Routing and Privacy Engine* supports the user in the definition of the privacy policies (with a *Privacy Policy Designer*) and based on these policies it interacts with the functional plugins to manage the request of subscription, to produce exports of clinical events and to answer to EHR explore request. The structure of the privacy policies is analogous to the ones in Definition 3.3.2 and they are defined by a data producer and used to serve a data consumer as described in Section 3.3.

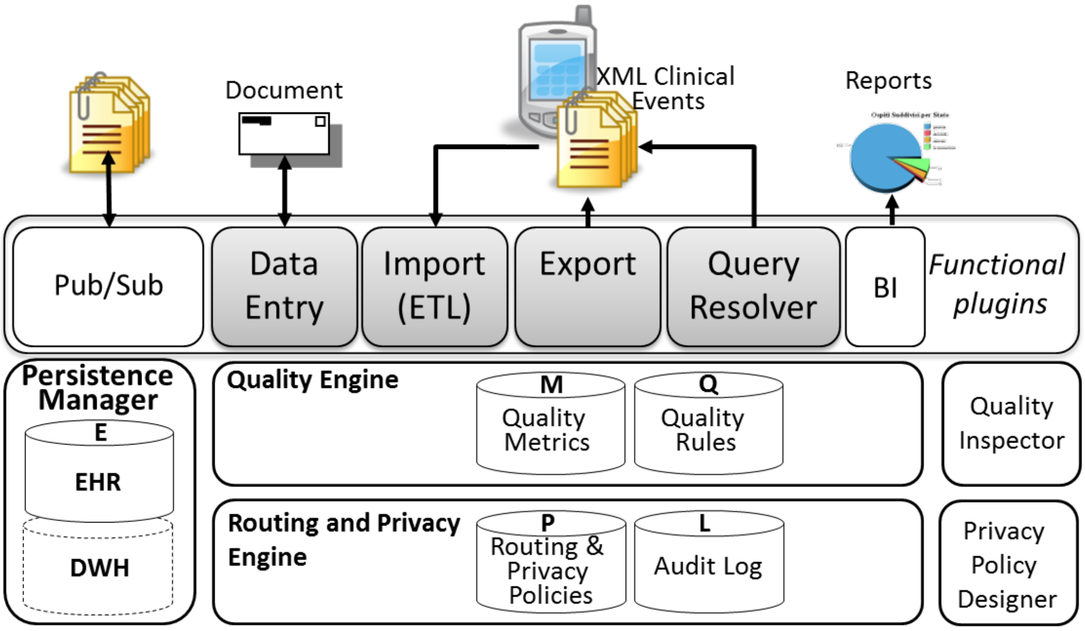


Figure 5.10: EHR modules at super-peer nodes. Grey modules are deployed also at the peers.

When a node is ready to share clinical events with other parties it publishes the list of sharable events to the publish/subscribe component of the super-peer with a description of their content so that a consumer node can subscribe to the categories of events it is interested in. The data producer can reject a request of subscription or accept it defining the specific privacy policies that restrict the access only to the parts of the events that are visible to the consumer for certain role and purposes (e.g. the general practitioner can access any sensitive data of the patient for healthcare purposes that instead a volunteer cannot see). The enforcement of the privacy policies in the super-peer node resolving the data access requests assures that only the data that is strictly necessary and authorized can flow.

The export component is used to “move” even big quantities of events to another node according to the privacy policies defined regulating the types of events that can be exported and their content. Similarly, the Query Resolver module for the query resolution and BI component for reporting can access only to the events and content allowed for the role and purpose of the consumer node.

The data *Quality Engine* management layer maintains a set of quality metrics to evaluate the level of quality of events and of quality rules to correct anomalies. The *Quality Inspector* interface allows users to see the quality of the events contained in the EHR and to ask for adjustments. The next section analyses more deeply the data quality management approach.

Data Quality Management

As for the EHR in Section 3.1 integration is achieved by sharing events where an *event* is a set of values $e_i = \{v_1, \ldots, v_n\}$ defined on the attributes $\mathbb{A}(e_i) = \{a_1, \ldots, a_m\}$ where $\Delta(a_i)$ is the domain of attribute a_i.
An *Event Export* X is a set of events $X = \{e_1, \ldots, e_n\}$. An event export is *privacy safe* with respect to a policy P defined on a set of attributes $\mathbb{A}(P) = \{a_1, \ldots, a_N\}$ iff $\mathbb{A}(X) \subseteq \mathbb{A}(P)$ where $\mathbb{A}(X) = \bigcup_i \mathbb{A}(e_i)$ for $i = 1, .., n$ and we denote it as $X \models P$. Intuitively X should contain only authorized events, that is, events with fields covered by some privacy rule.
As in Definition 3.3.2 in Section 3.3, a privacy rule specifies which fields are accessible to which role and purpose of use. We will not deal into details about privacy policy and their enforcement as the approach is analogous to the one presented in Section 3.4. Instead, in this section we focus more on another important aspect to consider in designing an EHR for developing countries that is the quality of the data delivered to the consumers. Often data is collected by poorly trained people, in precarious working conditions and in a hurry so that it is very easy to make mistakes. Delivering incomplete, wrong or outdated data to the governing bodies may induce them to take wrong decisions on critical medical and social processes. For this reason, data quality tracing and maintenance is as important as privacy management.
In literature, we can find a plethora of definitions of data quality reflecting the purpose of use of the data, the techniques available to measure and to directly correct the data anomalies or the processes producing the data [31]. We refer to the data quality definitions in [75] and in particular to the dimensions: completeness, consistency and timeliness. In particular, we adapted the definitions in [31] to the specific characteristics of the healthcare scenario:

1. *accuracy*: a clinical event is accurate if it does not contain typos.

2. *completeness*: refers to what extent the EHR represents the healthcare history of a patient. The goal is to collect as much information as possible on an individual in terms of clinical events reducing the amount of missing data (i.e. how many events and details have been lost).

3. *consistency*: the EHR should maintain correct information on the healthcare profile of patients based on constraints on the domain of attributes and their dependencies. The goal is to detect problems as early as possible (e.g. at the data entry phase).

4. *timeliness*: the EHR should be up-to-date to reflect the profile of patients.

Table 5.5: Sample Quality Metrics.

Dimension	Quality Metric
completeness	Domain: $\Delta(Drug)$=$\{aspirin, antivirus\}$
consistency	Integrity: $\Phi(DeadDate) > \Psi(BornDate)$
timeliness	Ordering: if $T(e_1) < T(e_2)$ i.e. if e_1 happens before e_2

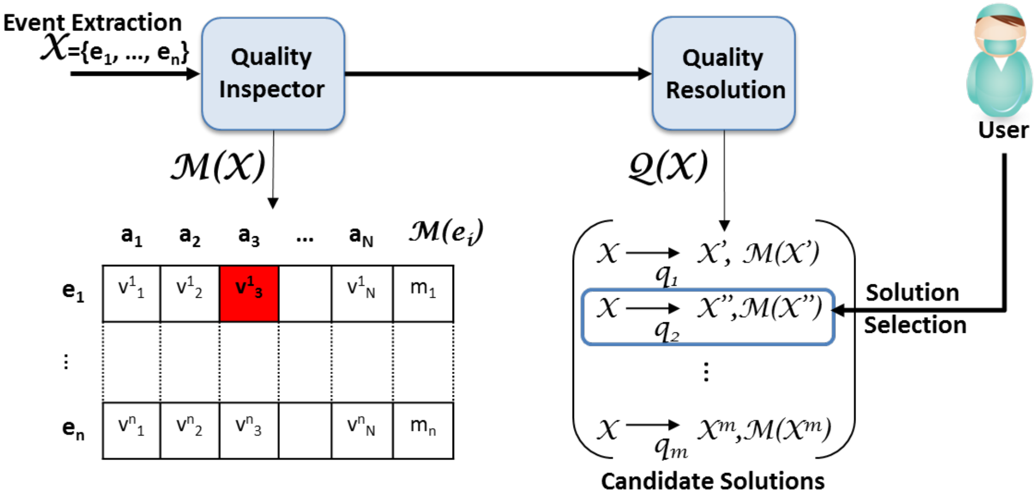

Event Extraction
$\mathcal{X}$={e_1, ..., e_n}
Quality Inspector
Quality Resolution
User
$\mathcal{M}(\mathcal{X})$
$\mathcal{Q}(\mathcal{X})$
a_1 a_2 a_3 ... a_N $\mathcal{M}(e_i)$
e_1 v^1_1 v^1_2 v^1_3 v^1_N m_1
⋮
e_n v^n_1 v^n_2 v^n_3 v^n_N m_n
$\mathcal{X} \xrightarrow{q_1} \mathcal{X}', \mathcal{M}(\mathcal{X}')$
$\mathcal{X} \xrightarrow{q_2} \mathcal{X}'', \mathcal{M}(\mathcal{X}'')$
⋮
$\mathcal{X} \xrightarrow{q_m} \mathcal{X}^m, \mathcal{M}(\mathcal{X}^m)$
Candidate Solutions
Solution Selection

Figure 5.11: Quality improvement with user feedback.

A quality metric gives the quality level of an event with respect to some dimensions. In Table 5.5 are reported some examples of quality metrics and the dimensions they refer to. Depending on the type of data included in the events and the kind of usage employed, different quality metrics could be defined.

Figure 5.11 shows the phases performed by the quality engine layer. Given a set of events $X = \{e_1, \ldots, e_n\}$, a quality inspector module applies the quality metrics to the events, $M(X)$, to get the level of quality of each event and to identify its anomalies (e.g. the value for a drug that is not in the allowed domain due to typos in the data entry phase). The engine can apply different quality resolution strategies $q_1, \ldots, q_m \in Q$ which transforms the input events to improve their quality level. Different resolution strategies can be considered from the more conservative one (do nothing approach), to the more destructive (delete the event). The quality engine cannot decide autonomously which quality resolution strategy is the best for the given event extraction as it depends on the purpose of use of the data and, consequently, on the decision of the user. So it proposes the different alternatives to the user with an explanation of the improvement of the quality level and loss of data that may be introduced especially if the "delete event" option is applied. Note that, as the system runs, it is possible to derive and use some predefined rules (e.g. apply always the same quality resolution strategy for certain types of events and purposes of usage).

The quality resolution operations could be applied locally to the given event extraction or it could be propagated to other nodes that may have received incorrect data. When a data warehouse is available in the node, it is possible to propagate the changes also to the data already loaded to correct the resulting BI analysis.

This approach poses also some research questions, like for example: how to find, in a reasonable time, the best quality resolution options minimizing the effort of the user in selecting the ones that better suits her needs and minimizing the loss of data.

5.2.3 Lessons Learned

We showed how the interoperability solution based on EDA presented in Section 3.1 [18], can be adapted to create an EHR in Mozambique. We showed that there are many issues to deal with and in particular: how to achieve data integration with limited resources and connectivity, how to guarantee that patient's privacy is protected, how to make users aware of the quality of the data they share and how to let them improve the overall quality of the data.

This section shows a solution for the first two research problems and an initial idea on the data quality management. It proposes an integration solution based on events with fine grained control on the privacy and quality of the data. Each node of the infrastructure can behave autonomously to create its local EHR repository of clinical events. This makes the infrastructure robust to unavailability of the internet connection or failures of the other nodes. But nodes can also interact with each other to get or share data by disseminating events to interested nodes.

The privacy of patients is preserved with a fine grained access control mechanism controlling how events and their content is disseminated and accessed. In addition, a quality engine layer allows to monitor, control and improve the quality of the events and to guide the user in the resolution of common errors (e.g. duplicates or typos).

An event is generated in correspondence to the steps of business processes executed across the different nodes. By tracing the events related to a certain individual, it is possible to reconstruct her healthcare profile and also the business process from which the events originate from. In this way, it is possible to verify that certain SLAs (Service Level Agreement) on the healthcare services provided are satisfied (e. g. the time elapsed between the entrance of a patient in a hospital and her examination cannot be more than 3 hours).

From our experience in the design and development of *SIS-H* [20] as an interoperability solution for developing countries, we saw that the feasibility of such projects depends not only on technical and organizational barriers but also economical. For that reason the *SIS-H* system was implemented with open source technologies that limits the costs and facilitates the use of the solution in public organizations (like the ministry of health). The same choices should be adopted also to develop the EHR. The infrastructure can be implemented in Java with Hibernate [1] for persistence management to support any DBMS (PostgreSQL [2], Apache Derby [78] and MySQL[3] are all good database options). For the reporting functionality of the business intelligence module a valid open source solution could be Jasper software.
The plug-in based architecture allows the use of different configurations (DBMS, importing/exporting functionalities or reporting tools) depending on the node. Typically, small health centres will benefit only of the data entry and event notification module and they may even avoid persisting the events. More complex and big nodes, like hospitals or the ministry of health, will deploy the whole infrastructure to perform more advanced analysis on the data collected.
The results expected from the introduction of an EHR in this context are:

- an increase of the EHR coverage: by collecting easily more data with clinical events, each node will create incrementally the profile of the patient;
- an improvement in the EHR quality: users can recognize problems in the data earlier and correct them autonomously with no involvement of the developers;
- access control and auditing capabilities: sensitive information usage is controlled to grant patients privacy and to understand who is responsible for privacy breaches;
- facilitate users work: it reduces the effort and time required for manual data entry with intuitive interfaces that will substitute gradually the paper, as data can flow at different levels of the healthcare organization in electronic form with no need to do repeated data entry operations to insert the data from paper to information systems;
- robust and flexible integration architecture: it is easier to share data among the nodes that can survive also autonomously in case there are no other peers available to share data (either because they do not agree to share events or because the internet connectivity is unavailable).

The solution is devised specifically for the healthcare domain but it can be applied also in other contexts like education or financial management and in any field in which

[3] http://www.mysql.com/

privacy and data quality are critical like education or food and agriculture management. The applications in the healthcare domain are diverse, for example:

- patient identification and family healthcare history management, by merging events of related individuals to discover duplicates and to follow the clinical history of a person among different structures and also to keep track of related people (e.g. mother-child relationship to reconstruct the family structure).
- BI for health monitoring and drugs consumption, to identify how epidemic events originate and spread on the territory and to keep the use and distribution of drugs under control. The same idea could be applied also to monitor food consumption and needs.
- definition and monitoring of prevention plans, by checking the compliance of a set of clinical events to the steps of a predefined plan (or in general business process) like for example the vaccination plans.

In all the application scenarios depicted above, it is important to retain control on the information shared with different organizations and to guarantee patients that their data is used according to the privacy policies defined.
Apart from privacy management, another benefit of our architecture is the capability to monitor and improve the quality of the events produced or events collected by a third party. In fact, data errors like misspells in the names of patients are very common in these scenarios resulting in duplicate data entries or in lost data as it cannot be associated to the right individual. The data quality support allows to identify these errors and to resolve them even before the data is used for BI tasks.

Chapter 6

Conclusion

6.1 Key Contributions

In this thesis work we analyzed privacy from two main perspectives: the first was the perspective of the IT designer required to design and develop an Electronic Health Record (EHR) compliant to the privacy regulations in different contexts; the second was the perspective of the privacy expert using the architectural solutions proposed in daily work. We provide solutions for the IT designer to realise an EHR with a privacy-aware design and for the privacy expert to define the privacy constraints with minimal effort. In this chapter, we summarize the contributions of the dissertations and highlights some directions for future work.

6.1.1 Architecture for EHR

The key innovation of the EHR solution proposed in this thesis work is the combination of the advantages of an Event-Driven SOA Architecture, providing decoupling, high reactivity to changes in the environment and capability to transmit such changes to many and different data consumers, with a tight control of the sensitive data. The architecture proposed satisfies the recent privacy regulations in the healthcare domain imposing restrictions on the processing modality including storage and communication of the data. The problem is already challenging and when multiple organizations in a rapidly evolving environment with different systems, privacy regulations and best practices are involved, it becomes even more complex. Our data integration solution follows a process- and event-based approach which makes it easy for new partners to come on-board, minimizes the development and maintenance effort required for the integration, and – perhaps most importantly – blurs the distinction between a data and a service integration project providing institutions with the benefits of both. This solves problems that more monolithic

integration strategies based on DWH experience in such context.
We showed how privacy and need for visibility can coexist thanks to an interaction protocol that meets regulatory requirements via a privacy aware event driven bus. Full control of the data is given to the institutions managing the data which are entitled to define fine-grained privacy policies regulating how data flows among the parties. The access to the data is performed only on-demand and for specific purposes captured by privacy policies defined directly by the data controllers. The enforcement of the privacy policies is performed in the bus without exposing any sensitive information unless authorized by the data controller. We proposed a two-phase communication protocol which departs from the classical pub/sub approach as additional information on an event is accessed only on-demand given a limited set of privacy-safe information distributed to the subscribers. The architecture and the prototype implementation has been validate with real health and welfare services deployed in the autonomous province of Trento showing, in this way, the applicability and worth of the proposed solution. In doing that we faced also organizational and legal problems which required us a systematic way to describe and motivate our design choices with respect to the privacy regulations.
Finally, we tried to adapt the solution to the context of developing countries in which a different organizational and technological environment requires to adjust certain assumptions to make the architecture more light and flexible.

6.1.2 Privacy constraints elicitation and specification

We have considered the problem of defining privacy constraints from a different perspective that common privacy policy definition tools. Our target user is the privacy expert elected in the data controller organization to manage the personal information. As such she will typically show good knowledge in the domain and in the privacy regulations but not necessarily the same familiarity with IT solutions. On the other end, an IT expert does not have the same knowledge of domain and legal context. This motivates us to study a privacy elicitation approach to exploit the knowledge and experience of the privacy experts in solving a problem for which there is not yet an automatic way to proceed: the definition of sensitive data. Our approach supports the user in understanding the privacy violations from the data to be protected and in "explaining" them in a form that the user can easily express and our algorithm can translate into privacy constraints which are directly enforceable on the underlying database. We ask to the user just to point out privacy violations and then our algorithm tries to guess the reason of the violation. This approach may draw the wrong conclusions but it guarantees the effect on the table analyzed by the user are exactly what she expects.

6.2 Future Work

We are planning to further extend the solutions proposed in this thesis by investigating more on some open problems and by relaxing some assumptions. Our goal is to provide a comprehensive solution for managing all the data lifecycle respecting various privacy preserving requirements, which is usable in real cases to relieve the user of unnecessary effort. In this section we introduce some of the research problems on which we will focus our future work.

6.2.1 The role of "roles"

Regarding the EHR architecture it should be investigated more the form of privacy constraints that are more suitable in a distributed EHR scenario. In the solution presented in this thesis work we assumed to have actors, roles and purposes to restrict the access to the event details. However, from the on-field experiments performed so far, we saw that the roles are difficult to identify by the users and also difficult to manage in a distributed environment as they require an agreement among the different institutions that is hard to create and to maintain. Roles should be an internal matter of each institutions instead of an element managed centrally by the data processor as in the architecture proposed. This is required also to be more adherent to the real healthcare scenario in which roles change frequently and dynamically. In contrast purposes are already a matter standardize at national level by the privacy guarantor office and can be easily included into contractual agreement defined statically (at design time) among the parties.
Another interesting evolution of the system is the involvement of data subjects (like patients and citizens of our case studies) in the data management process not only as providers of data but also as providers of privacy preferences. The collection of privacy policies from them is another interesting challenge as they may be poor both on the knowledge of the privacy regulations and on the IT world. This means we need to devise even more intuitive way to interact with them in collecting the privacy preferences and in explaining them how the system manages their personal information.

6.2.2 Event structure design

The structure of the events and the criteria used to divide the data into notifications and details has not followed a systematic approach but basically the desiderata of the users. In the future we want to define sound criteria to split the data into events minimizing the risk of privacy violations. The first strategy will be to apply the privacy constraints sampling approach to identify the sensitive data in our database at the data producer. Based on that information we can divide the data into events avoiding the violation of

privacy constraints.
A further step will be to identify privacy violations given by the combination of different events. Given the structure of the events and of the privacy policies it should be possible to identify the set of events that can be shared to a single consumer with no risk of violating any privacy policy.

6.2.3 Data and privacy constraints evolution

Our solution is strictly bound to the data shown to the user as the privacy constraints derived are valid for the specific database instance shown to the user. We cannot guarantee that the same constraints holds also in another database even with the same structure. On one side, this is a major benefit of our approach, as it relieves the user from knowing the underlying data structure but it requires to look only at the data. On the other side it represents also a weakness of the approach especially when the data already analysed by the user are modified or when new information arrives. We want to study strategies to manage data evolutions minimizing the effort of the user (repeating the entire sampling process is unfeasible) and minimizing the re-computation tasks. In a sense the user is telling us much more than what we are actually using in our elicitation process as we are capturing only the violations notified by the user. However, the user is indicating also the non-violations and this information is as important as the violations because it can drastically reduce the space of possible solutions to explore in figuring out the minimal form of the constraints. In the future we want to explore this problem in greater detail and to analyse open issues more deeply.

6.2.4 Outlier coverage and sample dimension

In our experiments we notice that there are many parameters to tune to improve the coverage and effectiveness of the algorithm in discovering new privacy constraints. The dimensions of the sample should not only consider usability requirements but also the actual distribution of the data. Our approach does not guarantee that no data is lost due to the sampling process and in particular data with low frequency are likely to be excluded from the samples even if they may represent important privacy violations. In fact, it may be very easy to identify uniquely an individual with data assuming quite rare information. Playing just with the dimensions of the sample may mitigate the problem but does not solve completely the issue. A better sampling heuristic should be devised taking more into account the data distribution and also the behaviour of the user in selecting more frequent vs less frequent information.

6.2.5 Conflict resolution

We assumed the user is coherent in indicating the privacy violations in different samples and we also assumed to have one single privacy expert to interact with. However, it is more realistic in an actual scenario to have different users providing feedback on the same database. This may lead to conflicts when the different feedback collected are combined. As in classical data quality and data cleaning procedure a fully automated approach to solve such anomalies is not possible but a strategy to discover, rank and propose to the users these conflicts should be provided.

6.2.6 Performance improvement

The last open problem of our elicitation approach is represented by the performances of the algorithms. We pushed a lot to find a solution that is as good as the optimal one. But this has a price in terms of time which may limit the applicability of our solution to small sets of data. In the future we plan to work both on improving the implementation applying more sophisticated optimization strategies and on devising less accurate, though acceptable, heuristics to make even big data sets affordable also for a real time interaction with the user.

Bibliography

[1] *Hibernate Reference Documentation. Relational Persistence for Idiomatic Java.*

[2] *PostgreSQL 9.1.2 Documentation.*

[3] COSO - The Committee of Sponsoring Organizations of the Treadway Commission-Internal Control - Integrated Framework, May 1994.

[4] Sarbanes-Oxley Act of 2002. http://www.soxlaw.com/, July 30 2002.

[5] *Oracle Database Security Guide 11g Release 1 (11.1)*, 2011.

[6] Delibera n. 3634 proposta da MAGNANI che disciplina le UVM, 29 December 2000.

[7] 104th Congress. 1st Session. HIPAA Health Insurance Portability and Accountability Act of 1996, 1996.

[8] Charu C. Aggarwal and Philip S. Yu. *Privacy-Preserving Data Mining: Models and Algorithms.* Springer Publishing Company, Incorporated, 2008.

[9] R. Agrawal, C. Johnson, J. Kiernan, and F. Leymann. Taming Compliance with Sarbanes-Oxley Internal Controls Using Database Technology. In *Data Engineering, 2006. ICDE '06. Proceedings of the 22nd International Conference on*, page 92, april 2006.

[10] Rakesh Agrawal, Tyrone Grandison, Christopher Johnson, and Jerry Kiernan. Enabling the 21st century health care information technology revolution. *Commun. ACM*, 50:34–42, February 2007.

[11] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. An XPath-based preference language for P3P. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 629–639, New York, NY, USA, 2003. ACM.

[12] AICPA and CICA. Generally Accepted Privacy Principles, GAPP. Technical report, Canadian Institute of Chartered Accountants (AICPA) and Canadian Institute of Chartered Accountants (CICA), August 2009.

[13] B.S. Alhaqbani. Privacy and trust management for electronic health records. *Science And Technology*, (June), 2010.

[14] Alon Halevy. Data Integration: a Status Report. Invited talk at the German Database Conference (BTW). Leipzig, Germany, 2003.

[15] Gustavo Alonso, Fabio Casati, Harumi A. Kuno, and Vijay Machiraju. *Web Services - Concepts, Architectures and Applications.* Data-Centric Systems and Applications. Springer, 2004.

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

[17] Annie I. Antón, Elisa Bertino, Ninghui Li, and Ting Yu. A roadmap for comprehensive online privacy policy management. *Commun. ACM*, 50:109–116, July 2007.

[18] Giampaolo Armellin, Dario Betti, Fabio Casati, Annamaria Chiasera, Gloria Martinez, and Jovan Stevovic. Privacy Preserving Event Driven Integration for Interoperating Social and Health Systems. In Willem Jonker and Milan Petkovic, editors, *Secure Data Management*, volume 6358 of *Lecture Notes in Computer Science*, pages 54–69. Springer Berlin / Heidelberg, 2010.

[19] Giampaolo Armellin, Dario Betti, Fabio Casati, Annamaria Chiasera, Gloria Martinez, Jovan Stevovic, and Tefo Toai. Event-Driven Privacy Aware Infrastructure for Social and Health Systems Interoperability: CSS Platform. In Paul Maglio, Mathias Weske, Jian Yang, and Marcelo Fantinato, editors, *Service-Oriented Computing*, volume 6470 of *Lecture Notes in Computer Science*, pages 708–710. Springer Berlin / Heidelberg, 2010.

[20] Giampaolo Armellin, Leandro Paolo Bogoni, Annamaria Chiasera, Tefo James Toai, and Gianpaolo Zanella. Enabling Business Intelligence Functions over Loosely Coupled Environment. *2nd International ICST Conference on e-Infrastructure and e-Services for Developing Countries (Africomm'10), Cape Town, South Africa*, 2010.

[21] Giampaolo Armellin, Annamaria Chiasera, Ganna Frankova, Liliana Pasquale, Francesco Torelli, and Gabriele Zacco. The eGovernment Use Case Scenario. In

Philipp Wieder, Joe M. Butler, Wolfgang Theilmann, and Ramin Yahyapour, editors, *Service Level Agreements for Cloud Computing*, pages 343–357. Springer New York, 2011.

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

[23] Paul Ashley, Satoshi Hada, Günter Karjoth, Calvin Powers, and Matthias Schunter. Enterprise Privacy Authorization Language (EPAL 1.2), 10 November 2003.

[24] Paolo Atzeni, Stefano Ceri, Stefano Paraboschi, and Riccardo Torlone. *Database Systems - Concepts, Languages and Architectures.* McGraw-Hill Book Company, 1999.

[25] Autonomous Province of Trento. Provincial law n. 13. http://www.consiglio.provincia.tn.it/documenti_pdf/clex_22190.pdf, 27 July 2007.

[26] Autonomous Province of Trento. Provincial law n. 6. http://www.consiglio.provincia.tn.it/documenti_pdf/clex_22004.pdf, 28 May 1998.

[27] B. C. M. Fung and K. Wang and R. Chen and P. S. Yu. Privacy-Preserving Data Publishing: A Survey of Recent Developments. *ACM Computing Surveys*, 42(4):14:1–14:53, June 2010.

[28] A. Barth, J. Mitchell, A. Datta, and S. Sundaram. Privacy and utility in business processes. In *Computer Security Foundations Symposium, 2007. CSF '07. 20th IEEE*, pages 279 –294, july 2007.

[29] Adam Barth and John C. Mitchell. Enterprise privacy promises and enforcement. In *Proceedings of the 2005 workshop on Issues in the theory of security*, WITS '05, pages 58–66, New York, NY, USA, 2005. ACM.

[30] Adam Barth, John C. Mitchell, and Justin Rosenstein. Conflict and combination in privacy policy languages. In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, WPES '04, pages 45–46, New York, NY, USA, 2004. ACM.

[31] Carlo Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino. Methodologies for data quality assessment and improvement. *ACM Comput. Surv.*, 41:16:1–16:52, July 2009.

[32] Kevin Beaver and Rebecca Herold. *The Practical Guide to HIPAA Privacy and Security Compliance.* Auerbach Publications, 2003.

[33] R. K. E. Bellamy, T. Erickson, B. Fuller, W. A. Kellogg, R. Rosenbaum, J. C. Thomas, and T. Vetting Wolf. Seeing is believing: Designing visualizations for managing risk and compliance. *IBM Systems Journal*, 46(2):205 –218, 2007.

[34] E. Bertino and R. Sandhu. Database security - concepts, approaches, and challenges. *IEEE Transactions on Dependable and Secure Computing*, 2(1):2 –19, jan.-march 2005.

[35] S.S. Bhowmick, L. Gruenwald, M. Iwaihara, and S. Chatvichienchai. Private-iye: A framework for privacy preserving data integration. In *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, page 91, 2006.

[36] Donelan K. Ferris T. Jha A. Kaushal R. Rao S. Rosenbaum S. Blumenthal D., DesRoches C. and Shield A. Health Information Technology in the United States, Where We Stand. Technical report, Massachusetts General Hospital and the Schol of Public Health and Health Services at George Washington University, June 2008.

[37] Kathryn Breininger and Mary McRae. ebXML Registry TC v3.0. Technical report, OASIS, 2005.

[38] Joe Butler, Juan Lambea, Michael Nolan, Wolfgang Theilmann, Francesco Torelli, Ramin Yahyapour, Annamaria Chiasera, and Marco Pistore. SLAs Empowering Services in the Future Internet. In John Domingue, Alex Galis, Anastasius Gavras, Theodore Zahariadis, Dave Lambert, Frances Cleary, Petros Daras, Srdjan Krco, Henning Mller, Man-Sze Li, Hans Schaffers, Volkmar Lotz, Federico Alvarez, Burkhard Stiller, Stamatis Karnouskos, Susanna Avessta, and Michael Nilsson, editors, *The Future Internet*, volume 6656 of *Lecture Notes in Computer Science*, pages 327–338. Springer Berlin / Heidelberg, 2011.

[39] Canada health infoway. http://www.infoway-inforoute.ca/.

[40] Marco Casassa Mont. Dealing with privacy obligations: Important aspects and technical approaches. In Sokratis Katsikas, Javier Lopez, and Gnther Pernul, editors, *Trust and Privacy in Digital Business*, volume 3184 of *Lecture Notes in Computer Science*, pages 120–131. Springer Berlin / Heidelberg, 2004. 10.1007/978-3-540-30079-3_13.

[41] Fabio Casati, Malu Castellanos, Umeshwar Dayal, and Norman Salazar. A generic solution for warehousing business process data. In *Proceedings of the 33rd inter-*

national conference on Very large data bases, VLDB '07, pages 1128–1137. VLDB Endowment, 2007.

[42] 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.

[43] Annamaria Chiasera and Fabio Casati. Gestione dei Dati in Amico. Technical report, Università degli Studi di Trento, 2009.

[44] Annamaria Chiasera, Fabio Casati, Florian Daniel, and Yannis Velegrakis. Engineering Privacy Requirements in Business Intelligence Applications. In *Proceedings of the 5th VLDB workshop on Secure Data Management*, SDM '08, pages 219–228, Berlin, Heidelberg, 2008. Springer-Verlag.

[45] Annamaria Chiasera, Tefo James Toai, Leandro Paolo Bogoni, Giampaolo Armellin, and Juan Jos Jara. Federated EHR: how to improve data quality maintaining privacy. *IST-Africa 2011, Gabarone, Botswana*, 2011.

[46] Michele Chinosi and Alberto Trombetta. Integrating Privacy Policies into Business Processes. *Journal of Research and Practice in Information Technology*, 41(2):155–170, 2009.

[47] Laura Chiticariu, Wang-Chiew Tan, and Gaurav Vijayvargiya. DBNotes: a post-it system for relational databases based on provenance. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, SIGMOD '05, pages 942–944, New York, NY, USA, 2005. ACM.

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

[49] CISIS. INF-3: Sistema federato di autenticazione. http://tinyurl.com/27yo92v.

[50] Chris Clifton, Murat Kantarcioğlu, AnHai Doan, Gunther Schadow, Jaideep Vaidya, Ahmed Elmagarmid, and Dan Suciu. Privacy-preserving data integration and sharing. In *Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, DMKD '04, pages 19–26, New York, NY, USA, 2004. ACM.

[51] 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 (CO-COA'10), HP Labs, Palo Alto, CA.*, 2010.

[52] Lorrie Cranor, Marc Langheinrich, and Massimo Marchiori. A P3P Preference Exchange Language 1.0 (APPEL1.0), 15 April 2002.

[53] Lorrie Cranor, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, and Joseph Reagle. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification W3C Recommendation 2002, 16 April 2002.

[54] Y. Cui and J. Widom. Lineage tracing for general data warehouse transformations. *The VLDB Journal*, 12:41–58, May 2003.

[55] Faiz Currim, Eunjin Jung, Xin Xiao, and Insoon Jo. Privacy policy enforcement for health information data access. In *Proceedings of the 1st ACM international workshop on Medical-grade wireless networks*, WiMD '09, pages 39–44, New York, NY, USA, 2009. ACM.

[56] 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.*, 5:169–202, May 2002.

[57] Data Protection Authority. Personal Data Protection Code, 2003.

[58] Data Protection Authority. Simplification of Notification Requirements and Forms, October 2008.

[59] Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Fragments and loose associations: respecting privacy in data publishing. *Proc. VLDB Endow.*, 3:1370–1381, September 2010.

[60] S. Dehousse, L. Liu, S. Faulkner, M. Kolp, and H. Mouratidis. Modeling delegation through an i*-based approach. In *Intelligent Agent Technology, 2006. IAT '06. IEEE/WIC/ACM International Conference on*, pages 393 –397, dec. 2006.

[61] Harald Deutsch and Fran Turisco. Accomplishing EHR/HIE (EHEALTH): lessons from Europe. Technical report, CSC, July 2009.

[62] A. Dogac, V. Bicer, and A. Okcan. Collaborative Business Process Support in IHE XDS through ebXML Business Processes. In *Data Engineering, 2006. ICDE '06. Proceedings of the 22nd International Conference on*, page 91, april 2006.

[63] Asuman Dogac, Gokce 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. *IJMSO*, 1(1):21–36, 2006.

[64] Robert H Dolin, Liora Alschuler, Calvin Beebe, Paul V Biron, Sandra Lee Boyer, Daniel Essin, Elliot Kimber, Tom Lincoln, and John E Mattison. The HL7 Clinical Document Architecture. *Journal of the American Medical Informatics Association*, 8(6):552–569, 2001.

[65] Wenliang Du, Zhouxuan Teng, and Zutao Zhu. Privacy-maxent: integrating background knowledge in privacy quantification. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 459–472, New York, NY, USA, 2008. ACM.

[66] European Commission empirica, eHealth Strategies. European countries on their journey towards national ehealth infrastructures - final european progress report. Technical report, European Commission, DG Information Society and Media, ICT for Health Unit, January 2011.

[67] Opher Etzion and Peter Niblett. *Event Processing in Action.* Manning Publications, 2010.

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

[69] Benjamin Eze, Craig Kuziemsky, Liam Peyton, Grant Middleton, and Alain Mouttham. Policy-based data integration for e-health monitoring processes in a B2B environment: experiences from Canada. *J. Theor. Appl. Electron. Commer. Res.*, 5:56–70, April 2010.

[70] Federal Trade Commission Bureau of Consumer Protection Division of Financial Practices. *The Gramm-Leach-Bliley Act Privacy of Consumer Financial Information*, November 12, 1999.

[71] Federal Trade Commission (FTC). *COPPA, Children's Online Privacy Protection Act of 1998*, 1998.

[72] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. Proposed nist standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4:224–274, August 2001.

[73] Philip W.L. Fong. Relationship-based access control: protection model and policy language. In *Proceedings of the first ACM conference on Data and application security and privacy*, CODASPY '11, pages 191–202, New York, NY, USA, 2011. ACM.

[74] Philip W.L. Fong and Ida Siahaan. Relationship-based access control policies and their policy languages. In *Proceedings of the 16th ACM symposium on Access control models and technologies*, SACMAT '11, pages 51–60, New York, NY, USA, 2011. ACM.

[75] Maria Grazia Fugini, Barbara Pernici, and Filippo Ramoni. Quality analysis of composed services through fault injection. *Information Systems Frontiers*, 11:227–239, July 2009.

[76] Floris Geerts, Anastasios Kementsietsidis, and Diego Milano. iMONDRIAN: A Visual Tool to Annotate and Query Scientific Databases. In Yannis Ioannidis, Marc Scholl, Joachim Schmidt, Florian Matthes, Mike Hatzopoulos, Klemens Boehm, Alfons Kemper, Torsten Grust, and Christian Boehm, editors, *Advances in Database Technology - EDBT 2006*, volume 3896 of *Lecture Notes in Computer Science*, pages 1168–1171. Springer Berlin / Heidelberg, 2006. 10.1007/11687238_84.

[77] Guido Governatori, Zoran Milosevic, and Shazia Sadiq. Compliance checking between business processes and business contracts. In *Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference*, pages 221–232, Washington, DC, USA, 2006. IEEE Computer Society.

[78] The PostgreSQL Global Development Group. *Derby Reference Manual*, 2011.

[79] Stuart. Hagen, Peter. Richmond, and United States. *Evidence on the costs and benefits of health information technology [electronic resource]*. Congress of the U.S., Congressional Budget Office, [Washington, D.C.] :, 2008.

[80] Alon Y. Halevy, Naveen Ashish, Dina Bitton, Michael Carey, Denise Draper, Jeff Pollock, Arnon Rosenthal, and Vishal Sikka. Enterprise information integration: successes, challenges and controversies. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, SIGMOD '05, pages 778–787, New York, NY, USA, 2005. ACM.

[81] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Min. Knowl. Discov.*, 15:55–86, August 2007.

[82] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques, 2nd edition.* The Morgan Kaufmann Series in Data Management Systems, Jim Gray, 2006.

[83] Health and Human Services Dept. (U.S.). *The Research-Based Web Design Usability Guidelines.* U.S. Government, August 2006.

[84] M. Hellinger and S. Fingerhut. Business Activity Monitoring: EAI meets data warehousing. *EAI Journal*, July 2002.

[85] 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 Aff*, 24(5):1103–1117, September 2005.

[86] Gregor Hohpe and Bobby Woolf. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions.* Addison-Wesley Professional, 2003.

[87] IHE. IHE - Integrating the Healthcare Enterprise XDS Profile. http://www.ihe.net/profiles/.

[88] Italian Data Protection Authority. Guidelines on the Electronic Health Record and the Health File, 2009.

[89] Jeff Ullman. Information Integration Systems, Answering Queries from Views. Lecture Notes at Stanford University, 2003.

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

[91] Jing Jin, Gail-Joon Ahn, Hongxin Hu, Michael J. Covington, and Xinwen Zhang. Patient-centric authorization framework for sharing electronic health records. In *Proceedings of the 14th ACM symposium on Access control models and technologies*, SACMAT '09, pages 125–134, New York, NY, USA, 2009. ACM.

[92] Joint Commission International. Accreditation Standards for Hospitas Medication Management and Use (MMU), 1 January 2011.

[93] Joint Commission International. Sentinel Event Data Summary. http://www.jointcommission.org/sentinel_event_statistics_quarterly, 30 September 2011.

[94] Diane Jordan and John Evdemon. Web Services Business Process Execution Language, WSBPEL, Version 2.0, 2007.

[95] Ivan Jureta, Stéphane Faulkner, and Pierre-Yves Schobbens. Clear justification of modeling decisions for goal-oriented requirements engineering. *Requirements Engineering*, 13(2):87–115, 2008.

[96] Jochen K üster, Ksenia Ryndina, and Harald Gall. Generation of Business Process Models for Object Life Cycle Compliance. In Gustavo Alonso, Peter Dadam, and Michael Rosemann, editors, *Business Process Management*, volume 4714 of *Lecture Notes in Computer Science*, pages 165–181. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-75183-0_13.

[97] R. Kimball and J. Caserta. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data.* John Wiley & Sons, 2004.

[98] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling.* Wiley, 2002.

[99] 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 *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering*, ASE '07, pages 429–432, New York, NY, USA, 2007. ACM.

[100] Petra Knaup, Oliver Bott, Christian Kohl, Christian Lovis, and Sebastian Garde. Electronic patient records: moving from islands and bridges towards electronic health records for continuity of care. *IMIA Yearbook 2007: Biomedical Informatics for Sustainable Health Systems*, (1):34–46, 2007.

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

[102] M. Lenzerini. Data Integration: A Theoretical Perspective. pages 233–246, 2002.

[103] Haoyuan Li, Yi Wang, Dong Zhang, Ming Zhang, and Edward Y. Chang. Pfp: parallel fp-growth for query recommendation. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 107–114, New York, NY, USA, 2008. ACM.

[104] Jun Li, S. Singhal, R. Swaminathan, and A.H. Karp. Managing data retention policies at scale. In *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pages 57 –64, may 2011.

[105] Jun Li, B. Stephenson, H.R. Motahari-Nezhad, and S. Singhal. GEODAC: A Data Assurance Policy Specification and Enforcement Framework for Outsourced Services. *IEEE Transactions on Services Computing*, 4(4):340 –354, oct.-dec. 2011.

[106] Ninghui Li, Tiancheng Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106 –115, april 2007.

[107] Linda T. Kohn, Janet M. Corrigan, and Molla S. Donaldson, Editors; Committee on Quality of Health Care in America, Institute of Medicine. *To Err Is Human: Building a Safer Health System.* The National Academies Press, 2000.

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

[109] Ashwin Machanavajjhala and Johannes Gehrke. On the efficiency of checking perfect privacy. In *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '06, pages 163–172, New York, NY, USA, 2006. ACM.

[110] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1, March 2007.

[111] Flora Malamateniou and George Vassilacopoulos. Developing a virtual patient record using XML and web-based workflow technologies. *I. J. Medical Informatics*, 70(2-3):131–139, 2003.

[112] 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):25, 2007.

[113] Mariemma I. Yagüe. Survey on XML-Based Policy Languages for Open Environments. *Journal of Information Assurance and Security*, 1(1):11–20, March 2006.

[114] D.J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J.Y. Halpern. Worst-case background knowledge for privacy-preserving data publishing. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 126 –135, april 2007.

[115] Tim Mather, Subra Kumaraswamy, and Shahed Latif. *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*. O'Reilly Media, Inc., 2009.

[116] Brenda M. Michelson. Event-Driven Architecture Overview. Event-Driven SOA Is Just Part of the EDA Story. Technical report, Patricia Seybold Group, 2006.

[117] Amalia R. Miller and Catherine Tucker. Privacy protection and technology diffusion: The case of electronic medical records. *Manage. Sci.*, 55:1077–1093, July 2009.

[118] Tim Moses. eXtensible Access Control Markup Language TC v2.0 (XACML). Technical report, OASIS, 2005.

[119] Alain Mouttham, Liam Peyton, Ben Eze, and Abdulmotaleb Saddik. Event-Driven Data Integration for Personal Health Monitoring. *Journal of Emerging Technologies in Web Intelligence*, 1(2), 2009.

[120] Municipality of Trento. Regulations for the protection of personal data of the municipality of Trento. Available at http://www.comune.trento.it/ as 'Regolamento per la tutela della riservatezza dei dati personali del comune di Trento', 2007.

[121] Municipality of Trento. Operational guidelines to privacy. Available at http://www.comune.trento.it/ as 'Guida operativa alla privacy', September 2009.

[122] NEMA Standards Publication. Digital Imaging and Communications in Medicine (DICOM). Part 1: Introduction and Overview. Technical report, Electrical Manufacturers Association, 2004.

[123] Mehmet Ercan Nergiz, Maurizio Atzori, and Chris Clifton. Hiding the presence of individuals from shared databases. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pages 665–676, New York, NY, USA, 2007. ACM.

[124] NHS-UK. NHS Connecting for Health. http://www.connectingforhealth.nhs.uk/.

[125] Qun Ni, Elisa Bertino, Jorge Lobo, Carolyn Brodie, Clare-Marie Karat, John Karat, and Alberto Trombetta. Privacy-aware role-based access control. *ACM Trans. Inf. Syst. Secur.*, 13:24:1–24:31, July 2010.

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

[127] NicTiz. eHealth in the Netherlands Policies, developments and status of cross-enterprise information exchange in Dutch healthcare, June 2008.

[128] NICTIZ-AORTA. AORTA the Dutch national infrastructure.

[129] OASIS. ebXML Registry Services OASIS Standard, v3.0, May 2005.

[130] Object Management Group. Business Process Model and Notation (BPMN), 2009.

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

[132] Office of the National Coordinator for Health Information Technology. Maternal and child health ahic extension/gap. Technical report, U.S. Department of Health and Human Services Office of the National Coordinator for Health Information Technology, 2008.

[133] Basel Committee on Banking Supervision. Basel III: A global regulatory framework for more resilient banks and banking systems, June 2011.

[134] Committee on Data Standards for Patient Safety Board on Health Care Services. Key Capabilities of an Electronic Health Record System. Letter Report. Technical report, Institute of Medicine of the National Accademies Washington, 2003.

[135] Parliament and Council. Directive on protection of individuals with regard to the processing of personal data and on the free movement of such data, 1995.

[136] Parliament of United Kingdom. *Data Protection Act*, 1998.

[137] Vibhor Rastogi, Michael Hay, Gerome Miklau, and Dan Suciu. Relationship privacy: output perturbation for queries with joins. In *Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '09, pages 107–116, New York, NY, USA, 2009. ACM.

[138] Joseph Reagle and Lorrie Faith Cranor. The platform for privacy preferences. *Commun. ACM*, 42:48–55, February 1999.

[139] Shariq Rizvi, Alberto Mendelzon, S. Sudarshan, and Prasan Roy. Extending query rewriting techniques for fine-grained access control. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, SIGMOD '04, pages 551–562, New York, NY, USA, 2004. ACM.

[140] Shazia Wasim Sadiq, Guido Governatori, and Kioumars Namiri. Modeling control objectives for business process compliance. In *BPM'07*, pages 149–164, 2007.

[141] Alberto Siena, John Mylopoulos, Anna Perini, and Angelo Susi. Designing law-compliant software requirements. In *Conceptual Modeling - ER 2009*, pages 472–486, 2009.

[142] William W. Simons, Kenneth D. Mandl, and Isaac S. Kohane. The PING personally controlled electronic medical record system: Technical architecture. *Journal of the American Medical Informatics Association*, 12(1):47 – 54, 2005.

[143] Ian Sommerville. *Software engineering (7th ed.)*. Addison Wesley Longman Publishing Co., Inc., 2004.

[144] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):571–588, 2002.

[145] S Heard T Beale. *openEHR - Architecture Overview*. openEHR, April 2007.

[146] Wang Chiew Tan. Research problems in data provenance. *IEEE Data Eng. Bull.*, 27(4):45–52, 2004.

[147] Wang Chiew Tan. Provenance in Databases: Past, Current, and Future. *IEEE Data Eng. Bull.*, 30(4):3–12, 2007.

[148] P. C. Tang, J. S. Ash, D. W. Bates, J. M. Overhage, and D. Z. Sands. Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption. *Journal of the American Medical Informatics Association : JAMIA*, 13(2):121–126+, 2006.

[149] Integrating the Healthcare Enterprise. IHE IT Infrastructure (ITI) Technical Framework. Volume 1 (ITI TF-1) Integration Profiles. Revision 8.0 Final Text, August 19 2011.

[150] TSE, Tavolo di lavoro permanente Sanit Elettronica delle Regioni e delle Province Autonome. GdLT: IBSE. Strategia architetturale per la Sanit Elettronica. Technical report, Ministro per l'Innovazione e le Tecnologie, March 2006.

[151] Fatih Turkmen and Bruno Crispo. Performance evaluation of XACML PDP implementations. In *Proceedings of the 2008 ACM workshop on Secure web services*, SWS '08, pages 37–44, New York, NY, USA, 2008. ACM.

[152] Vassilios S. Verykios, Elisa Bertino, Igor Nai Fovino, Loredana Parasiliti Provenza, Yucel Saygin, and Yannis Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Rec.*, 33:50–57, March 2004.

[153] Lingyu Wang, Sushil Jajodia, and Duminda Wijesekera. Securing OLAP Data Cubes Against Privacy Breaches. *IEEE Symposium on Security and Privacy*, 0:161, 2004.

[154] David R R Webber. Understanding ebxml, UDDI, XML/EDI. Technical report, XML Global Technologies Inc., 2000.

[155] Jennifer Widom. Trio: a system for integrated management of data, accuracy, and lineage. In *Proceedings of the Second Biennial Conference on Innovative Data Systems Research (CIDR '05)*, 2005.

[156] Raymond Chi-Wing Wong, Jiuyong Li, Ada Wai-Chee Fu, and Ke Wang. (alpha, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. *In: Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 20-23 Aug 2006, Philadelphia, USA*, 2006.

[157] World Health Organization. ICD-10: International Statistical Classification of Diseases and Related, Health Problems. 10th Revision. Available at http://apps.who.int/classifications/icd10/browse/2010/en, 2010.

[158] Xiaokui Xiao and Yufei Tao. Anatomy: simple and effective privacy preservation. In *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pages 139–150. VLDB Endowment, 2006.

[159] Mariemma Yagüe. Survey on XML-Based Policy Languages for Open Environments. *Journal of Information Assurance and Security*, pages 11–20, 2006.

[160] Eric Siu-Kwong Yu. *Modelling strategic relationships for process reengineering.* PhD thesis, Toronto, Ont., Canada, Canada, 1996.

Appendix A

Socio-Healthcare Data Warehouse Design

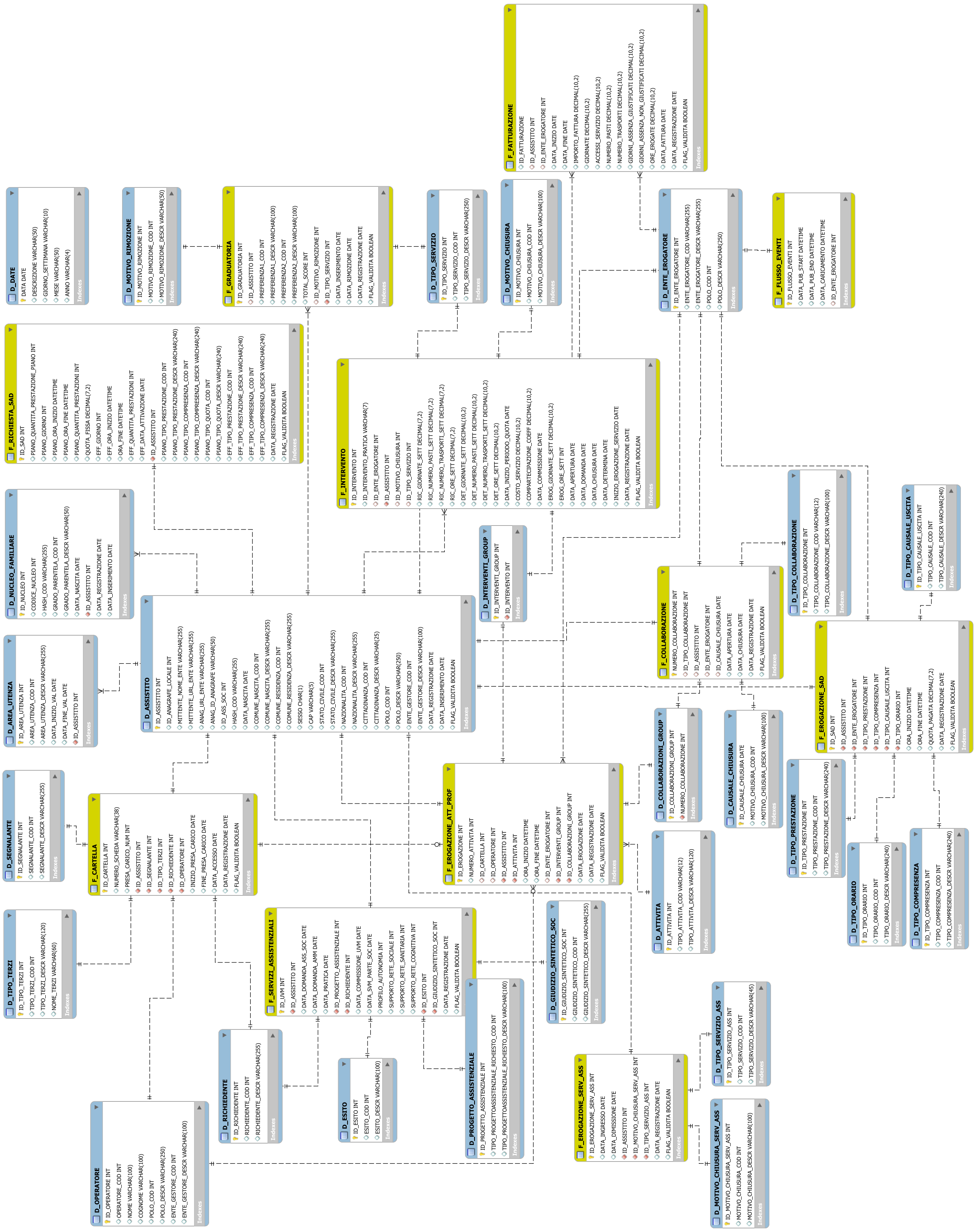


Figure A.1: Data structure design of DWH for the EHR of the CSS project in the Social and Healthcare domains.

Appendix B

FP-Tree Construction and Frequent Patterns Discovery

The following two algorithms are adapted from [82] and allow the creation of an FP-Tree from a set of tuples in a database D and the discovery of all the frequent patterns in D without candidate generation.

Algorithm 4: $FPTreeCreation(\mathcal{D}) \mapsto \mathbb{T}$.

```
Data: $\mathcal{D} = \{t_1, t_2, \dots, t_n\}$, $t_i = [A_1 = v_1, A_2 = v_2, \dots, A_k = v_k], i = 1, \dots, n$ transactions in table $R$
Result: FP-Tree $\mathbb{T}$ of the input transactions $\mathcal{D}$.
/* Compute frequency of atomic condition $A = v$ in transactions of $\mathcal{D}$ */
$L$ ←frequency of atomic conditions in $\mathcal{D}$ /* Header Table with frequencies and pointers to Tree nodes */
foreach ($t \in \mathcal{D}$) do
    foreach ($c \in t$) do
        if ($c \in L$) then
            $L[c]++$
        else
            $L[c] = 1$
        end
    end
end
$sort(L)$ /* Sort items in decreasing frequency */
$\mathbb{T} \leftarrow \perp$ /* Set the root of the FP-Tree */
foreach ($t \in \mathcal{D}$) do
    $sort(t)$ /* sort $t$ wrt $L$ */
    $c \leftarrow t[1]$ /* first item in transaction */
    $C \leftarrow t[2], \dots, t[k]$ /* remaining item in transaction */
    $insert(c, C, \mathbb{T})$ /* insert each transaction in $\mathbb{T}$ */
end
```

Algorithm 6 shows the FP-Growth procedure as implemented in [82]. It allows to generate the list of frequent patterns in the FP-Tree by invoking $FPGrowth(\mathbb{T}, null)$. The algorithm is slightly different as it does not apply any pruning on the minimum support.

Algorithm 5: *insert*$(c, C, \mathbb{T})$.

```
Data:
$c$ atomic condition
$C$ remaining part of the transaction
$\mathbb{T}$ FP-Tree
$N \leftarrow getChild(\mathbb{T}, c)$ /* get child node equal to $c$ */
if $(n = null)$ then
    $N \leftarrow newNode(c)$/* create a new node with condition $c$ */
    $addChild(\mathbb{T}, N)$ /* add a child node with support 1 */
else
    $increaseSupport(N)$ /* increment the support of the node */
end
if $(C \neq \perp)$ then
    $c \leftarrow C[1]$
    $C \leftarrow t[2], \ldots, t[k]$ /* remaining item in transaction */
    $insert(c, C, N)$ /* insert the remaining part of the transaction */
end
```

Algorithm 6: $FPGrowth(\mathbb{T}, \alpha)$.

```
Data:
$\mathbb{T}$ FP-Tree to mine
$\alpha$ pattern
if ($\mathbb{T}$ contains a single path $\mathcal{P}$) then
    foreach combination $\beta$ of nodes in $\mathcal{P}$ do
        generate pattern $\beta \cup \alpha$ with support = minimum support of nodes in $\beta$
    end
else
    foreach $a_i$ in Header Table $L$ of $\mathbb{T}$ do
        generate pattern $\beta = a_i \cup \alpha$ with support = $a_i$.support
        construct $\beta$'s conditional pattern base and then $\beta$'s conditional FP-Tree $\mathbb{T}_\beta$
        if $(\mathbb{T}_\beta \neq \Phi)$ then
            call $FPGrowth(\mathbb{T}_\beta, \beta)$
        end
    end
end
```

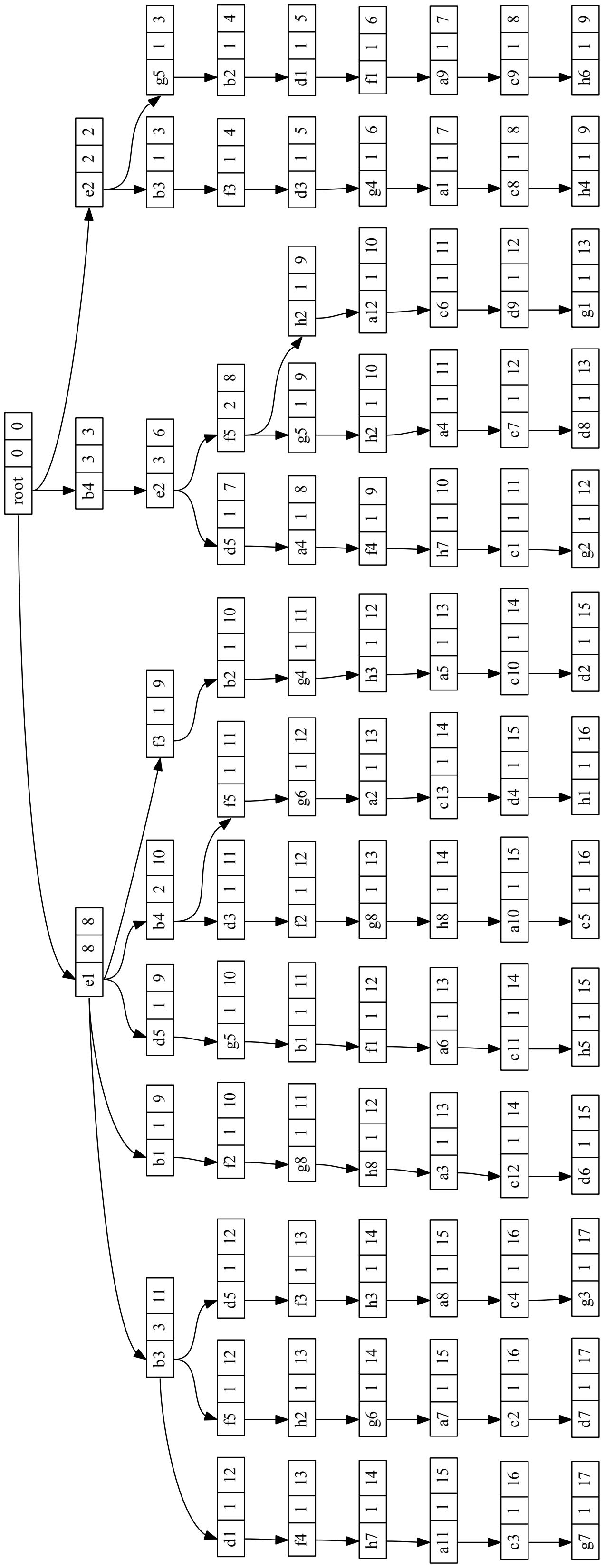


Figure B.1: FPTree from sample relation in Figure 4.5a.