A System to Handle Privacy Obligations in Enterprises♦

Marco Casassa Mont
Trusted Systems Laboratory
HP Laboratories Bristol
HPL-2005-180
October 21, 2005*

obligation management, privacy obligations, privacy policies, privacy enforcement, compliance monitoring, obligation management system, identity management, data governance, data lifecycle management

Privacy obligations dictate expectations and duties that need to be carried out by enterprises when storing, processing and disclosing personal data. Privacy obligations can be defined by data subjects, by laws and/or enterprises' internal guidelines. They require enterprises to deal with data governance and data lifecycle management activities, including data retention and deletion aspects, notifications, data transformation and execution of complex workflows.

The management and enforcement of privacy obligations is a challenging task: it involves legal, organizational, behavioral and technical aspects. It is still a green area open to research. Our goal is to introduce degrees of automation and a systemic approach to the problem in order to allow enterprises to reduce the involved costs and simplify their overall management process. This document (based on the author's MSc thesis on this topic) provides a detailed analysis of privacy obligations in an identity management context, within enterprises: it describes their core properties and highlights key requirements.

A model to represent, manage, enforce and monitor privacy obligations is introduced. In this model, obligations are "first class" entities, not subordinated to access control criteria. We compare it against related work and highlight its advantages. We describe the architecture of an obligation management system, based on this model: we also provide technical and implementation details about a working prototype, that has been implemented by HP Labs (in the context of the EU PRIME project) to demonstrate the feasibility of our approach.

Our obligation management system can be exploited right now by current, state-of-the-art, identity management solutions: in particular, we analyse how to achieve this in the context of user provisioning and account management. We describe how we have successfully integrated our obligation management system prototype with HP Select Identity (HP leading edge solution in the area of user provisioning and account management) by: enabling the definition of fine-grained privacy obligations on personal data when disclosing and provisioning this data; scheduling, enforcing and monitoring privacy obligations on personal data by leveraging HP Select Identity's web service APIs and its workflow capabilities.

The final part of this document discusses the results we have achieved so far, it describes a few open issues that must be addressed and introduces our next research activities, to be done in the context of HP Labs and EU PRIME project.

* Internal Accession Date Only
♦Universita'di Torino – Dipartimento de Informatica – http://www.di.unito.it, Turin, Italy Sept. 2005
Approved for External Publication
© Copyright 2005 Hewlett-Packard Development Company, L.P.

A System to Handle Privacy Obligations in Enterprises

Marco Casassa Mont
marco.casassa-mont@hp.com

Trusted Systems Laboratory
Hewlett-Packard Laboratories
Bristol, UK

Abstract

Privacy obligations dictate expectations and duties that need to be carried out by enterprises when storing, processing and disclosing personal data. Privacy obligations can be defined by data subjects, by laws and/or enterprises' internal guidelines. They require enterprises to deal with data governance and data lifecycle management activities, including data retention and deletion aspects, notifications, data transformation and execution of complex workflows.

The management and enforcement of privacy obligations is a challenging task: it involves legal, organizational, behavioral and technical aspects. It is still a green area open to research. Our goal is to introduce degrees of automation and a systemic approach to the problem in order to allow enterprises to reduce the involved costs and simplify their overall management process.

This document (based on the author's MSc thesis on this topic) provides a detailed analysis of privacy obligations in an identity management context, within enterprises: it describes their core properties and highlights key requirements.

A model to represent, manage, enforce and monitor privacy obligations is introduced. In this model, obligations are "first class" entities, not subordinated to access control criteria. We compare it against related work and highlight its advantages.

We describe the architecture of an obligation management system, based on this model: we also provide technical and implementation details about a working prototype, that has been implemented by HP Labs (in the context of the EU PRIME project) to demonstrate the feasibility of our approach.

Our obligation management system can be exploited right now by current, state-of-the-art, identity management solutions: in particular, we analyse how to achieve this in the context of user provisioning and account management. We describe how we have successfully integrated our obligation management system prototype with HP Select Identity (HP leading edge solution in the area of user provisioning and account management) by: (1) enabling the definition of fine-grained privacy obligations on personal data when disclosing and provisioning this data; (2) scheduling, enforcing and monitoring privacy obligations on personal data by leveraging HP Select Identity's web service APIs and its workflow capabilities.

The final part of this document discusses the results we have achieved so far, it describes a few open issues that must be addressed and introduces our next research activities, to be done in the context of HP Labs and EU PRIME project.

UNIVERSITA' DEGLI STUDI DI TORINO

Facoltà di Scienze M.F.N.

Corso di Laurea Magistrale in Sistemi per il Trattamento dell'Informazione

TESI DI LAUREA

A System to Handle Privacy Obligations in Enterprises

Candidato	**Relatore**	**Controrelatore**
Marco Casassa Mont	Prof.sa Maria Luisa Sapino	Prof. Franco Sirovich

Anno Accademico 2004/2005

Index

Figures

1 Introduction

This thesis describes research done at HP Labs during the last two years. The related projects and activities, described in this thesis, have been led by the author who contributed since their initial stages, both in terms of research and development. These projects have subsequently been extended in the context of an EU project and led to the development of a few prototypes.

The problem addressed by this thesis is in the area of privacy management: how to manage and enforce privacy obligations in an enterprise context.

Privacy is a fundamental right of human beings. This is recognised by laws and legislation in most countries of the world. These laws apply both to the real and the digital world. In particular, in the digital world, including the Internet and the web, personal data and digital identities are subject to data protection directives and regulations that dictate how these data can be collected, stored, processed, disclosed and retained, based on people's consent and their preferences.

Privacy is a complex topic: many definitions of privacy are available, but none of them fully captures all its properties and implications. At the very base, privacy is about:

- "The quality of being secluded from the presence or view of others";
- "The right of an individual to be secure from unauthorized disclosure of information about oneself that is contained in documents and digital data";
- "Ensuring that individuals maintain the right to control what information is collected about them and how it is used as well";
- "For citizens and consumers, freedom from unauthorized intrusion. For organizations, privacy involves the policies that determine what information is gathered, how it is used, and how customers are informed and involved in this process. Privacy is a legal issue, but it is also an information security issue".

The advent of the Internet and the digitalization of the communication media allowed people and organisations to easily store, process, analyse and exchange digital identities and personal data. On one hand, this enables cheaper, faster and more effective interactions and transactions: on the other hand misuses of large amounts of personal data could damage and have negative consequences both for the involved people and organisations.

Both data subjects (people) and organizations are affected by privacy matters: the former need to be assured that their identities and personal data are used for the agreed purposes, according to their preferences and consent; the latter need to ensure that they are compliant with privacy laws and related requirements imposed by customers.

In this thesis we focus on privacy and privacy management aspects, from an enterprise perspective. Enterprises see privacy management as an important aspect of identity management: in the last few years they have been heavily investing in identity management solutions to collect, store, access and process identity information and personal data of customers, employees and business partners. Being able to leverage these investments to handle privacy is a priority.

We want to provide tools and solutions to "good-willing" enterprises to allow them to be compliant with privacy policies and enforce data subjects' requirements and preferences.

Privacy management includes methodologies and technologies to process personal data and identities in a privacy compliant way consistently to regulations and data subjects' rights. This involves handling and enforcing privacy policies that can dictate recommendations on data subjects' rights, permissions and obligations that must to be satisfied.

We address the specific problem of managing and enforcing privacy obligations. The privacy obligation management area is a green field open to research and innovation.

Privacy obligations define and describe the expected behaviours and constraints to be satisfied by enterprises when handling confidential and personal data. They dictate how to handle personal data and deal with their lifecycle management in a privacy-aware way. This includes dealing with data retention, data deletion, periodic notifications and requests of authorizations, execution of complex privacy-aware workflows.

We analyse privacy obligations and describe our research and development work to build an obligation management system, as follows.

Chapter 2 introduces concepts and principles related to identity and privacy management: it also introduces the concept of privacy obligations in the context of privacy.

Chapter 3 describes in more details the addressed problem i.e. dealing with the management and enforcement of privacy obligations in enterprises.

Chapter 5 analyses related work and compares it against our suggested approach for dealing with privacy obligations.

The main contributions of our work are described in:

- Chapter 4, which contains our analysis of privacy obligations and their properties;
- Chapters 6 and 7, which describe our privacy obligation model, our obligation management framework along with the architecture of a system to manage, enforce and monitor privacy obligations. Chapter 7 also contains technical implementation details of an obligation management prototype developed in the context of the EU PRIME project;
- Chapter 8 describes a few scenarios where our privacy obligation management system can be deployed along with related use cases;
- Chapter 9 illustrates how our technical solution for managing privacy obligations can be deployed in a real-world enterprise's identity management solution, by leveraging and extending its user provisioning and account management capabilities. Details are provided about the integration of our privacy obligation management system with HP Select Identity, a state-of-the art identity management solution.

Chapter 10 contains a discussion of current results, open issues and plans for the future.

Chapter 11 draws a few conclusions.

As anticipated at the beginning of this chapter, the work described in this thesis has been done in the context of a research and development project at HP Labs, Bristol, UK [Hewl05b].

This work has subsequently been extended and carried on in the context of the EU PRIME project [Prim05], as part of an international research effort to address the problem of dealing with Privacy for Identity Management in Europe.

Specifically, the implementation of a more advanced version of the prototype has happened in the context of the EU PRIME project by the same HP project team.

2 Identity and Privacy Management

This chapter introduces relevant terminology and concepts about identity and privacy management.

As highlighted in the introduction we will focus on an enterprise context: the aim is to help enterprises to manage digital identities and personal data by satisfying privacy regulations and data subjects' expectations.

This chapter also provides more details about the concept of privacy obligation and how it fits in the context of privacy management.

Dealing with the management and enforcement of privacy obligations within enterprises is the key problem addressed in this thesis and the focus of the next chapters.

2.1 Identity Management

Identity management is an important aspect for enterprises, e-commerce and government to underpin their business processes and services and enable digital interactions and transactions.

There are different competing demands on what identity management should provide, concerns on what it should focus on and conflicting interests: enterprise focus vs. consumer focus, mobility vs. centralisation, legislation vs. self-regulation, subjects' control vs. organisations' control, privacy vs. free market, etc. They are dictated by various stakeholders, including data/identity subjects (people), enterprises, service providers and government agencies, which have different objectives and priorities when dealing with identity management.

Many products and solutions are available on the market: they address problems in different areas such as provisioning and accounting, authentication, authorization and data consolidation. Currently, they are evolving, towards their consolidation and integration with the IT stack (i.e., networks, platforms, OSs, applications, middleware, services, etc.) and the associated business solutions; nevertheless most of these products and solutions still manage identity aspects in relatively static, closed and well-controlled environments.

Identity management has strong links with the management of security, trust and privacy: all these aspects are directly or indirectly involved when managing identity information. In today's products, there is little integration and synergy with these management aspects. Each product usually provides its own set of management tools. Because of this fragmentation, any change of or request to enforce new requirements on identity information might need a lot of work and take long time to be achieved.

Current trends [CaBP03] suggest that the digital world is going to be more and more flexible and dynamic. Barriers and boundaries between enterprises, organizations and government agencies are getting increasingly indistinct as people cover multiple roles and are involved in activities that span across heterogeneous environments. This creates a broad new set of opportunities in the personal, social and business areas. On the other hand this also creates new threats and issues.

Digital identities and identity management play a strategic role in enabling this new world and addressing the related issues [CaBP03].

2.1.1 Aspects of Identity

Identity and *identity management* are overloaded terms. They are used in different contexts, at different levels of abstractions, with different meanings. This chapter introduces some terminology and discusses a few identity-related aspects.

Entities in the physical and digital world (i.e., people, devices, systems, services, etc.) can be intrinsically characterised and described by means of *attribute*s. We will also use the "*data subjects"* term to refer to these entities. Some of their attributes, including personal details, financial information, social information, etc., can be used for identification and profiling purposes.

In this thesis we refer to *identity information* as a set of attributes (along with their values) describing relevant aspects of an entity [PaVi01]. We will use, in an interchangeable way, the terms *identity information* and *personal data*.

This information is dynamic: the set of attributes and their values can change over time.

Different *views* on an entity's identity information can be created, disclosed, accessed and used by multiple parties. A *view* consists of an aggregation (i.e. a set) of one or more attributes. Each attribute can assume different values, depending on the view it belongs to and the context where it is used.

A *digital identity* (or *identity*) is itself a *view* on the identity information associated to an entity, at a specific point of time. A digital identity, of course, has additional properties such as its uniqueness in a specified context. Digital certificates, credentials, etc., are examples of digital identities.

In general, *views* on identity information might include any meaningful aggregations of attributes that can be used for identification and profiling purposes, including e-mail addresses, credit card details, personal information, roles, rights, etc.

Attributes and views can be qualified by *metadata*, i.e., additional attributes such as information about their certifier(s), their provenance and validity, management policies, etc. Metadata might define *relationships, references* and *dependencies* among attributes and views.

For simplicity (unless otherwise stated) we will use in an interchangeable way the terms "digital identity", "personal information", "personal data" and "view on identity information". We stress the fact that a "digital identity" might consist of any aggregation of attributes and not only classic attributes, such as the ones defined by X.509 identity certificates [HFPS99].

Today, digital identities are mainly associated to people. In the future their usage will be increasingly extended to devices, trusted systems, web services and any type of proxies and agents that mediate interactions and transactions in the digital world.

Figure 1 shows the relationships between identity information, attributes, views and usage contexts. In general "identity subjects" are aware of the existence of only a part of their "identity information", they "own" just a portion of it and they can directly control only a subset of it.

Broadly speaking, rather than talking of "data owners" it is more appropriate to talk about "data subjects" as identity information is not necessarily owned by the entity this information refers to. From an identity subject's point of view, there are multiple perceptions of their identity information [Pato03], [CBG+02]:

- "**Me Me**": it is the part of identity information that the subject is aware of and directly controls;
- "**Known Me**": it is the part of identity information that the subject is aware of and indirectly controls;
- "**Unknown Me**": it is the part of identity information that the subject is not aware of and has no control on.

Multiple views can exist on an entity's identity information. These views can be used within and across different contexts (personal, social, e-commerce, government, business, etc.) to enable interactions and transactions.

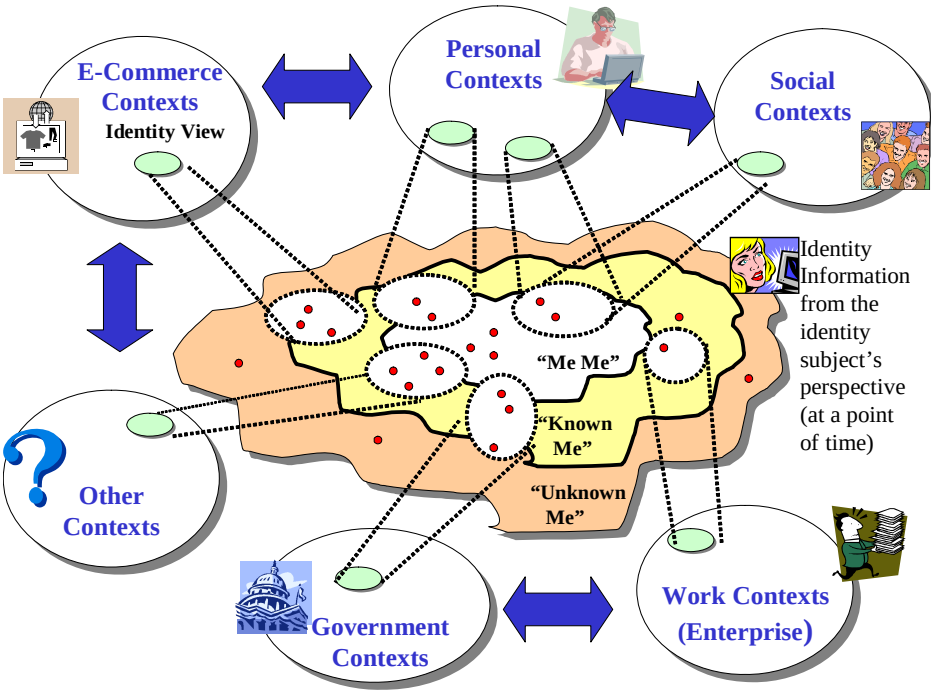


Figure 1: Identity Information, Views and Context

The management of an entity's identity information is constrained and characterised by important aspects, including:

- **Control**: different stakeholders can access, use and manage this identity information and/or related views. These stakeholders include the data subject, *third parties* that are known by the subject (such as certification authorities, authorised e-commerce sites, trusted third parties/TTPs, etc.) and *unknown third parties* (such as credit rating agencies, identity thieves, etc.).
- **Contexts***: identity information* and *identities* can be disclosed, accessed and used by different stakeholders in one or more contexts, including personal, social, e-commerce, enterprise and government ones. This can happen via a variety of means and systems including personal appliances, enterprise systems and web services.
- **Time**: identity information changes over time. New attributes are created, others are updated and others again are not valid anymore. The management of these changes is fundamental as it directly affects identity's integrity and consistency, its trustworthiness and its privacy and, indirectly, authentication, authorization, access control, etc.

Figure 2 and 3 are an attempt to graphically represent the above three aspects and highlight their relationships with identity information:

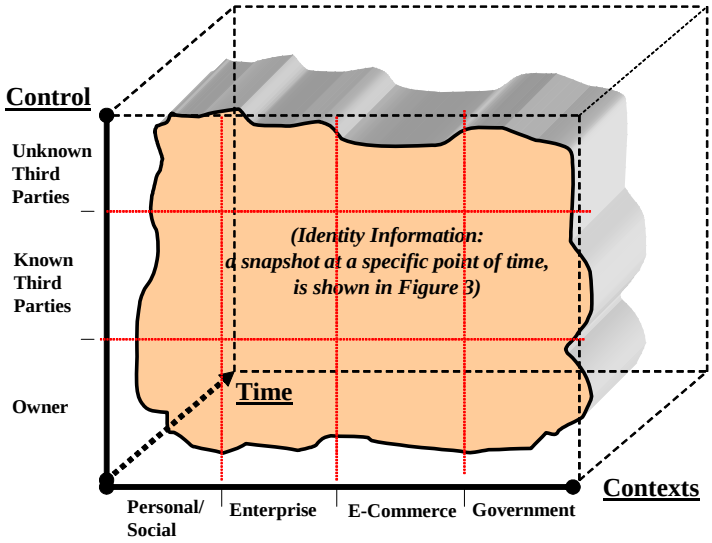


Figure 2: Multi-dimensional Aspects of Identity Information

Figure 2 conveys a (simplified) multi-dimensional perspective on identity information, i.e., who controls which kind of identity information, at a specific point of time. Not only identity information changes over time but also the contexts where it is used and the stakeholders that control it change too: changes to identity information might happen in different contexts and be driven by different stakeholders.

Figure 3 provides more details about a snapshot of an entity's identity information at a specific point of time:

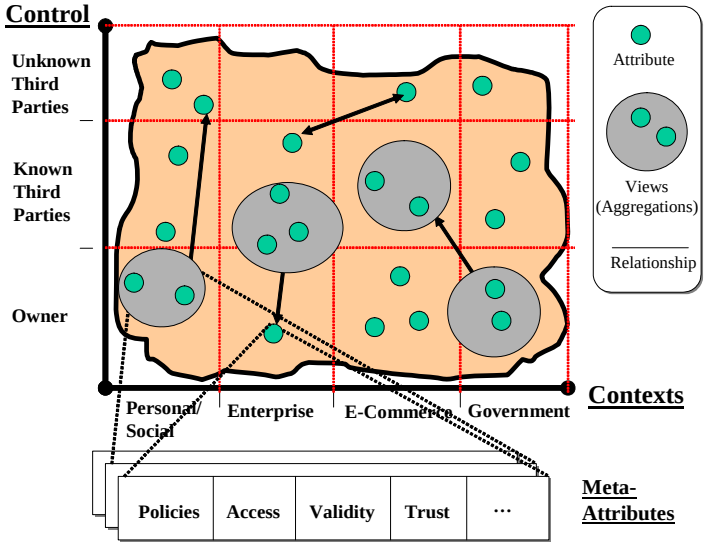


Figure 3: Attribute Aggregations, Relationships and Meta-attributes

As anticipated, identity information is made of attributes, views and relationships. They are qualified by metadata, including management (business, security, privacy, trust, etc.) policies, access constraints, validity, etc. All these aspects can change over time.

Identity management has to deal with the management of this information along with its metadata (meta-attributes), cope with changes and make sure that the associated policies are satisfied.

The next section of this chapter describes aspects of the current identity management landscape including current solutions and related issues.

2.1.2 Identity Management Landscape

The current identity management landscape is very complex because of the multiple interests, perspectives, concerns and technologies that are involved.

As anticipated in the introduction of this chapter, there are different competing aspects on what identity management should provide and concerns on what it should focus on. These conflicting interests include: enterprise focus vs. consumer focus, mobility vs. centralisation, legislation vs. self-regulation, subjects' control vs. organisations' control, privacy vs. free market – see figure 4:

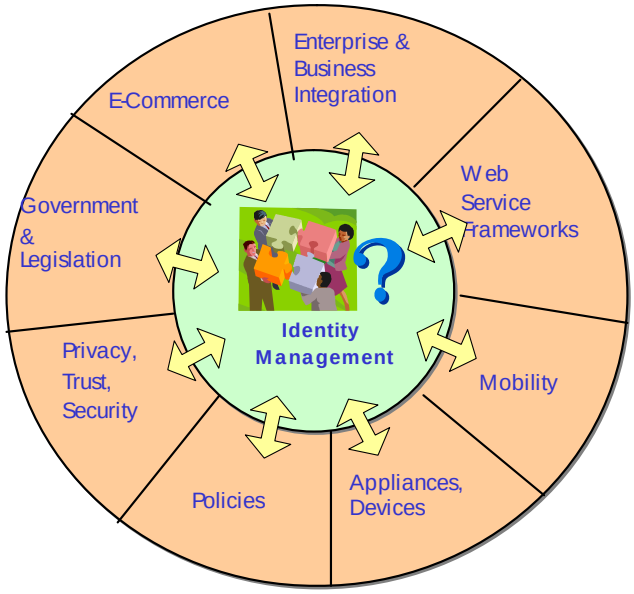


Figure 4: Identity Management Landscape

Priorities, interests and perspectives on identity management differ, depending on the involved stakeholders:

- Enterprises are driven by their business objectives and needs. They aim at the management of large sets of identity and personal data to enable their businesses, rationalize their assets and simplify business interactions with partners and customers, manage the information lifecycle of their workforce and deal with access management to enterprise resources;
- E-commerce sites and service providers manage consumers' identity information with the hope to increase their sales, understand customers' needs, customize the provision of services or just sell this information to third parties;

- Government agencies are concerned with the control and protection of personal information of their citizens, the provision of strong and undeniable authentication mechanisms and the automation/rationalisation of the provision of their services via the web and the Internet;
- People have different concerns and needs depending on the role they play: they are right in the middle (or, depending on the point of view, the source) of most of the above competing aspects. As employees or consumers, they want to access and use services in the simplest and more efficient way, without any hassle. As private citizens they might be concerned about their privacy, have a lack of trust on institutions, demand for more accountability of the involved parties.

This variety of interests, concerns, along with new emerging technologies, contributes to increase the complexity of identity management.

All these aspects influence each other, via a spiral of potentially conflicting requirements. For example, new legislations are addressing citizens' needs for privacy and, on the other hand, they are constraining the way enterprises, e-commerce sites and service providers deal with the processing of personal information. The mobility of employees creates on one hand security and trust management problems to enterprises and organisations, on the other hand new business opportunities. Last but not least, emerging appliances and web service frameworks create new issues such as dealing with the identities of devices and web services and coping with delegation aspects and trust matters.

From a technological and IT perspective, identity management is just one of the aspects that are involved in the management of business solutions and the overall IT stack (i.e., networks, platforms, OSs, applications, middleware, services, etc.). Figure 5 shows some of the elements that influence it.

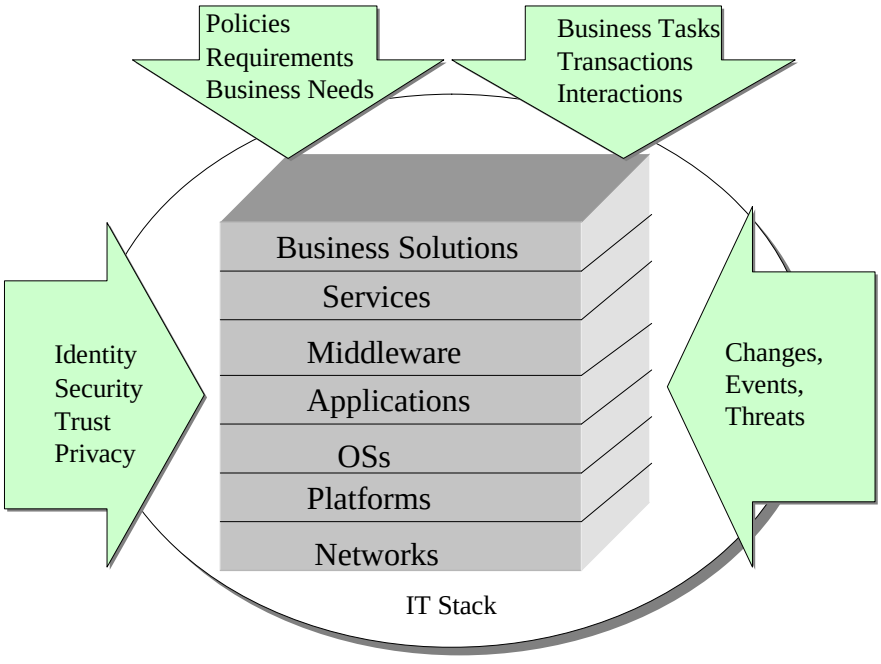


Figure 5: Context where Identity Management Operates

Identity management must be considered in a holistic way by including (among other things) the management of security, trust and privacy along with the management of policies, requirements and changes. All these aspects are very inter-related and affect business solutions and the IT stack at different levels of abstraction. Of course, the context dictates which IT elements and which identity management aspects are meaningful.

Further complexity derives from the fact that the execution of business tasks or the management of digital interactions and transactions can span among multiple domains. For example, in an e-commerce context, a digital transaction might require the involvement of identity e-commerce sites and the exchange of identity information among these sites: this has strong implications in terms of management of trust, privacy, authentication, authorization and accountability. Similarly this is true for B2B interactions or transactions within supply-chain communities.

The effectiveness and validity of identity management products and solutions depends, among other things, on how good they are at keeping identity information in a consistent and up-to-date state, satisfy related management policies and legal requirements, preserve privacy and trust and ensure that security requirements are fulfilled.

New requirements, new policies, changes or threats might affect the configuration of elements in the IT infrastructure and business solutions. As a consequence, complex reconfiguration activities might need to be done on multiple components, at different levels of abstraction.

Identity management plays a key role in this space: identity aspects need to be managed rapidly and orchestrated with security, trust and privacy aspects. In environments where business and customers' needs change frequently, identity management solutions have to be flexible and adaptable.

2.1.3 Current Identity Management Technologies and Solutions

This chapter provides an overview of the state of the art of identity management products and solutions and discusses a few related issues. A more detailed analysis can be found in [Gabl02], [Gaw01], [Senf03].

Today, many identity management products and solutions are available on the market. They supply functionalities such as authentication, SSO, authorization, auditing, provisioning, data storage, links to legacy systems and data consolidation. They target different types of users and contexts including e-commerce, service providers, enterprises and government institutions.

Figure 6 shows the main components and functionalities provided by current identity management products and solutions:

- **Directory services, meta-directories, virtual directories and databases** deal with the representation, storage and management of identity and profiling information and provide standard APIs and protocols for their access [Penn02a], [Neue02]. In particular, meta-directories address the important problem (especially for large organizations and enterprises) of consolidating, integrating and preserving the consistency of data, disseminated in a variety of heterogeneous systems, geographically spread across organization sites.
- **Authentication, authorization and auditing** are core identity management functionalities. Authentication, in particular, is provided in a variety of ways ranging from local authentication on a system to complex distributed authentication [Smit01],

[Burt02], including single-sign-on (SSO) within and across organizational boundaries [Volc01], [Decl02]. Recent initiatives, including Liberty Alliance Project [LiAP05a], [LiAP05b], aim at the provision of SSO for a federated environment [Blum02], by leveraging identity providers acting as trusted third parties. Similarly, authorization functionalities are provided in a variety of forms, usually coupled with auditing capabilities. Authorization can include simple access control management at the OS level, more sophisticated role-based access control - RBAC [FeKu92] - up to flexible, distributed, policy-driven authorization, at the application and service levels.

- **Provisioning and longevity** solutions [Penn02b] are used by enterprises, organizations and e-commerce sites to deal with the lifecycle management of identities, including the enrolment, customization, modification and destruction of accounts associated to users, employees and customers along with associated identity information (including rights, permissions and access control information). Related functionalities deal with the issuance, certification, management and revocation of digital entitlements and credentials in a secure and trusted way. In particular PKI-based solutions [HFPS99] are available for this purpose but their adoption is not so widespread, especially in inter-organisational contexts, because of the intrinsic trust management problems, the complexity of CA hierarchies and related costs.
- **Self Service, Personalization and Single-Sign-On** components provide core functionalities to end-users (data subjects) in terms of self-registration and management of their personal information and identities along with mechanisms for single-sign-on across multiple systems and services (within and across organisational boundaries).

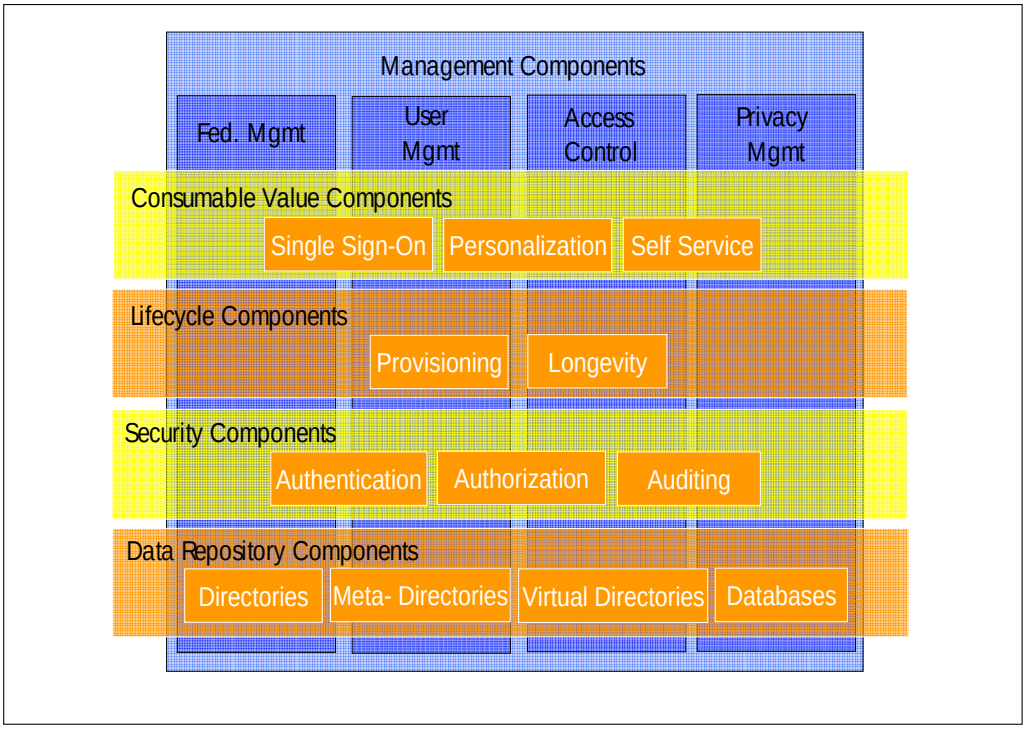


Figure 6: Current Identity Management Solution Stack

These components provide core identity management functionalities in the following areas:

- **User management:** management of the lifecycle of user accounts associated to data subjects, within organisations;
- **Access control management:** management of access rights and permissions associated to users within organisations;
- **Federated identity management**: management of identity information, access rights and permissions across organisational boundaries;
- **Privacy management:** management of identity information in a way that is compliant to data subjects' requirements, laws and organisational guidelines.

The above components and solutions have mainly been described from an enterprise and organisational perspective: this is where identity management solution providers are concentrating most of their efforts and where, currently, most of the money is. Nevertheless, as we anticipated, identity management is much more that this and involves other stakeholders.

Identity management technologies include, among many other things:

- **authentication devices**: smartcards, biometric devices, authentication tokens, etc.;
- **anonymity services**;
- **cryptography schemas** based on the RSA public/private key paradigm or on alternative schemas, such as IBE [BoFr01], [Cock01], [CHM+02];
- **trusted platforms** [TCPA01], [Pear02], [Micro05];
- **emerging standards** [Blum02] including:
  - signed and encrypted XML [W3C03a], [W3C03b];
  - XACML [OASI05a];
  - XKMS [W3C01];
  - SAML [OASI05b];
  - SOAP [W3C03c];

2.1.4 Open Issues

Important open issues that need to be addressed in the identity management area are:

- **Identity thefts and identity-based frauds**. Internet identity thefts and related frauds [Arno00], [CADT00] are fast growing crimes, because of poor security and privacy practices and the underestimation of the involved risks. In the future, when digital identities and profiles are going to be more pervasive and used for day-by-day life tasks, the consequences of those crimes could affect very seriously people's lives and businesses. Identity management solutions need to play a key role in protecting identities and profiles, help organisations to enforce good management practices and, in case of thefts and frauds, help to detect the criminals or support forensic analysis;
- **Lack of control on identity information.** Data subjects have little control over the management of their identity information. It is very hard (if not impossible) for the subjects of identity information to define their own privacy policies (or delegate this task to trusted third parties), check for their enforcement, track in real-time the dissemination and usage of their personal information, be alerted when there are attempts

to use or misuse it, etc. Because of emerging data protection laws, new legislations and the need of service providers to simplify the overall management, there is a tendency towards the delegation to users of the authoring of their identity profiles. Despite this, identity management solutions mainly address the needs and requirements of the "consumers" of identity information, not their subjects. Identity management solutions need to evolve and include mechanisms that allow people to author their management policies and monitor their enforcement [CaPB03] (or delegate these activities to trusted third parties). Identity management solutions will have to quickly adapt to changes dictated by people's requirements and needs;

- **Accountability** is an important issue for identity management. There is currently a lack of mechanisms and solutions to ensure accountability when dealing with the management of identity information. Today, when people disclose their identity information to third parties, they rely on them to protect and manage this information, as agreed. It is a matter of trust. Unfortunately, the number of cases where identity information is leaked or misused is increasing, due to lack of security, incompetence or fraudulent behaviours. On the other hand, solutions are required to help organisations to demonstrate that they acted honestly and with due-diligence whilst dealing with personal data. Identity management solutions need to provide strong, undeniable auditing and logging mechanisms and solutions that can be flexibly configured based on policies [CaPB03], [BaFS03]. In doing this they might need to leverage trusted platforms and rely on trusted third parties;
- **Complexity** of identity management solutions: it is a barrier for common people and, increasingly, also for administrators, given the broad set of skills and knowledge that are required to have to make them work. New privacy and data protection laws, the increasing awareness of people about their rights, the need of organisations and service providers to adapt to customers' requirements and the consequent workload for organisations, might be important factors to move towards delegation and the provision of simpler to use identity management solutions;
- **Privacy** is an important issue that has to be addressed directly by identity management products and solutions. There are increasing concerns about the fact that enterprises, e-commerce sites, governments and third parties can access and correlate people's identity information, sell this information or misuse it. Laws and legislation only partially address the problem. Despite the fact that many efforts have been made at the legislation level, there are still a lot of problems to be addressed. Privacy laws can differ quite substantially depending on national and geographical aspects. The enforcement of privacy policies is a key requirement [BBC+03]. It has strong implications and repercussions on identity management, especially in contexts where identity information is disclosed during interactions and transactions involving multiple third parties. This includes multiparty B2B communities (such as supply-chains) and federated e-commerce sites. From an enterprise and organisational perspective, this creates the problem of how to defend their reputation and brand when things go wrong. Mechanisms and solutions are required to help them to demonstrate that they acted honestly and with due diligence whilst dealing with personal data. Identity management solutions need to provide accountable mechanisms to interpret and enforce privacy policies customized by the identity subjects, delegate the management to third parties trusted by the identity subjects, adapt to changes in privacy legislation and quickly deal with threats that could compromise the confidentiality of personal data.

All these issues are under research by the identity management community in a variety of contexts. In particular privacy management is perceived as being a hot topic for enterprises because of more and more stringent regulatory compliance requirements and an increased awareness of people and consumers about their rights.

Our work specifically focuses on privacy management aspects and addresses related problems in the area of management and enforcement of privacy obligations on personal data/identities: details about the addressed problems are provided in the next chapters.

The next section of this chapter provides more details about privacy management and how privacy obligations fit in this context.

2.2 Privacy Management

Privacy management is an important issue for identity management: it has implications on storing, handling, processing, exchanging, disclosing and dealing with the lifecycle management of identities and personal data, in a way that is compliant to and consistent with data subjects' expectations, laws and privacy guidelines.

Privacy management is a wide topic and it spans across a variety of contexts - user side, enterprise, e-commerce and government - where personal information and identity are collected and used.

We specifically focus on privacy management for enterprises. This is an important context as enterprises collect and process large amounts of personal information and digital identities: hence they are subject to pressure both from citizens and laws to handle that information in an honest and law-abiding way. We analyse some core privacy concepts and principles defined by current laws and legislation. We then discuss the implications for enterprises.

2.2.1 Privacy Laws

A lot of work has been done in terms of privacy legislation often driven by local or geographical needs or to enable business interactions and exchanges of personal information across nations. This includes:

- **European Community data protection** Directive [EuCo05]: The data protection Directive applies to 'any operation or set of operations which is performed upon personal data,' called 'processing' of data. Such operations include the collection of personal data, its storage, disclosure, etc. The Directive applies to data processed by automated means (e.g. a computer database of customers) and to data that are part of or intended to be part of non automated 'filing systems' in which they are accessible according to specific criteria (for example, the traditional paper files, such as a card file with details of clients ordered according to the alphabetic order of the names);
- **Various US laws**, addressing legislative aspects of privacy in specific areas:
    - **HIPAA** [Hipa05]: Health Insurance Portability and Accountability Act (HIPAA) responds to concerns from citizens, the health care industry and government agencies for enhanced security and privacy of individual health information. Furthermore, HIPAA creates uniform methods to bill and share health information electronically between healthcare providers, payers and other organizations involved with healthcare delivery and payment;

- **COPPA** [Copp00]: The Children's Online Privacy Protection Act (COPPA), effective April, 2000, applies to the online collection of personal information from children under 13. Among other things, rules spell out what a Web site operator must include in a privacy policy, when and how to seek verifiable consent from a parent and what responsibilities an operator has to protect children's privacy and safety online;
- **GLB** [GLB03]: The Gramm-Leach-Bliley Act (GLB) has privacy provisions relating to consumers' financial information. Under these provisions, financial institutions have restrictions on when they may disclose a consumer's personal financial information to non affiliated third parties. Financial institutions are required to provide notices to their customers about their information-collection and information-sharing practices. Consumers may decide to "opt out" if they do not want their information shared with non affiliated third parties. The GLB Act provides specific exceptions under which a financial institution may share customer information with a third party and the consumer may not opt out. All financial institutions are required to provide consumers with a notice and opt-out opportunity before they may disclose information to non affiliated third parties outside of what is permitted under the exceptions.
- **SOX** [SOX05]: The SOX financial reporting legislation, intended to create reform and restore investor trust in public companies, makes senior managers directly accountable for the design and effectiveness of internal controls and the accuracy and integrity of financial reporting. Since controls are only as effective as the people accountable for the process, these new rules dramatically impact a company's need to ensure that all employees understand their role in enhancing and maintaining the controls. SOX legislation requires collecting and storing information about access control, identities and log information to be checked for compliance purposes. These activities have privacy implications, in terms of handling the collected personal and confidential data in an appropriate way;
- Other US laws having an impact on privacy, such as PATRIOT ACT, etc. [PrLa05];

- **Safe Harbour** [Safe00]: The European Commission’s Directive on Data Protection went into effect in October, 1998, and would prohibit the transfer of personal data to non-European Union nations that do not meet the European "adequacy" standard for privacy protection. While the United States and the European Union share the goal of enhancing privacy protection for their citizens, the United States takes a different approach to privacy from that taken by the European Union. The United States uses a sector-oriented approach that relies on a mix of legislation, regulation, and self regulation. The European Union, however, relies on comprehensive legislation that, for example, requires creation of government data protection agencies, registration of data bases with those agencies, and in some instances prior approval before personal data processing may begin. As a result of these different privacy approaches, the Directive could have significantly hampered the ability of U.S. companies to engage in many trans-Atlantic transactions. In order to bridge these different privacy approaches and provide a streamlined means for U.S. organizations to comply with the Directive, the U.S. Department of Commerce in consultation with the European Commission developed a "safe harbour" framework. The safe harbour -- approved by the EU in 2000-- is

an important way for U.S. companies to avoid experiencing interruptions in their business dealings with the EU or facing prosecution by European authorities under European privacy laws. Certifying to the safe harbour will assure that EU organizations know that your company provides "adequate" privacy protection, as defined by the Directive.

- **Specific national privacy initiatives, around the world** [Laur04].

Guidelines are also available on the protection of privacy and flows of personal data, including OECD guidelines [Oecd80], that describe concepts such as collection limitation, data quality and purpose specification principles and online privacy policies [Priv04].

Large enterprises that are geographically distributed across different nations might need to comply with different privacy laws.

2.2.2 Privacy Policies and Related Views

Privacy policies can be used to represent and describe privacy laws, guidelines and privacy statements. They are usually expressed in natural language that needs to be interpreted and understood by people.

Two related perspectives/views can be used to analyse the aspects and implications of privacy policies:

1. Policies can be seen as a collection of constraints and requirements on **how to handle personal data**, based on stated purposes for which these data have been collected, consent (e.g. opt-in, opt-out options on particular matters) given by data subjects, limitations on collection, usage, disclosure and retention of data – see Figure 7;
2. Policies can also be seen as a collection of constraints and requirements expressing **rights, permissions and obligations** – see Figure 8: rights of data subjects, permissions given by data subjects to data receivers and obligations that data receivers must fulfil when handling personal data.

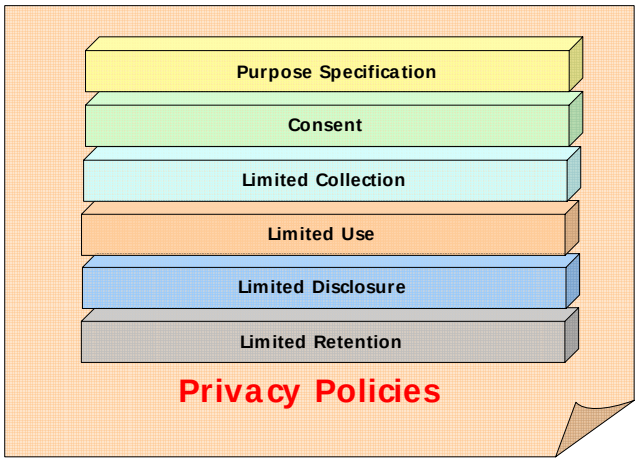


Figure 7: 1st View of Privacy Policies

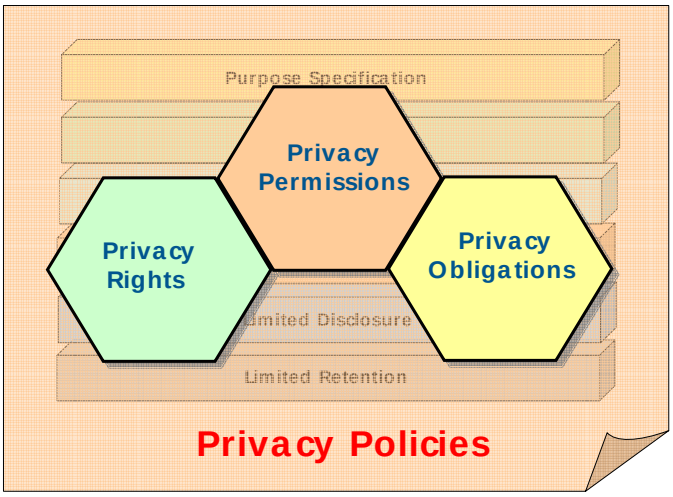


Figure 8: 2nd View of Privacy Policies

These two views are basically equivalent in terms of the types of privacy policies they can cover and describe.

In this thesis we will mainly consider the second view, where privacy policies are seen in terms of rights, permissions and obligations.

Specifically, we will focus on technical aspects related to the management and enforcement of privacy obligations as part of the wider problem of dealing with privacy policies.

2.2.3 Privacy Management by Enterprises

Enterprises store, manage and process large amounts of personal and confidential data related to their employees, customers and partners.

As previously described, on one hand this information is fundamental to enable their business processes, interactions and transactions. For example, to enable e-commerce transactions, employees' processes or government interactions, data subjects (end-users) are required to disclose part of their personal data, such as identity, financial details or medical information.

On the other hand, personal data should be accessed and used only for the purposes for which they have been disclosed according to the consent of the data subjects and existing obligations.

Enterprises increasingly recognise that dealing correctly and honestly with privacy matters can have beneficial returns for their businesses.

Figure 9 summarises the main impacts and consequences that privacy has on enterprises. Enterprises are subject to a lot of pressure dictated by:

- **Regulatory compliance**: laws and legislations impose good behaviours and potentially complex processes that enterprises must be aware of and compliant with;
- **Customers' needs and requirements**: customers are more and more aware of their rights. Being able to satisfy their needs in terms of available privacy choices and deal with the fulfilment of related promises can be quite challenging for enterprises;

- **Internal guidelines imposing good practices**: medium and large enterprises can also define internal guidelines and policies dictating good practices and approved processes that need to be fulfilled by employees and business partners.

The rewards for being privacy-aware can include beneficial returns for enterprises, not only in terms of being compliant with laws but also in terms of branding, trust, customers' satisfaction and additional business opportunities.

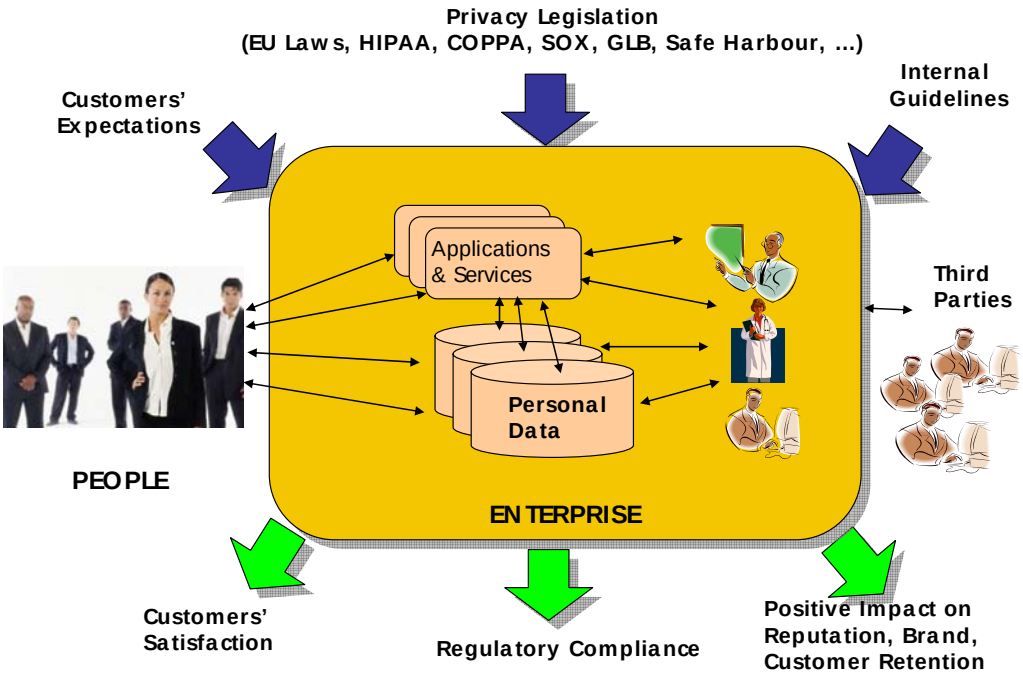


Figure 9: Privacy-related Aspects affecting Enterprises

The overall process of managing data and related policies, including privacy policies, is commonly referred as "data governance".

Figure 10 shows the main steps involved in a data governance process. At the core of this process there are people, personal data and identity information and the overall set of systems, applications and services that access and process these data.

The set of steps involved in the data governance process include:

- **Policy Development and Modeling**: this step consists in understanding which relevant policies should apply to the given context, model them and refine in a way that can be operationally managed and enforced;
- **Data Inventory**: this step consists in creating and maintaining an inventory of relevant data (personal information, digital identities, confidential data, etc.) stored and handled by the enterprise that are subject to any of the policies analysed in the previous step;

- **Gap and Risk Analysis**: this step involves the analysis of gaps and risks (for example from a business/security/privacy perspective), given the context (people, roles, systems, applications, services, interactions, etc.) and the definition of appropriate strategies and tactics to handle and manage these policies;
- **Policy Deployment**: this step requires the deployment of refined policies to the components that are in charge of handling and making decisions based on them;
- **Policy Enforcement**: this steps involved the enforcement of these policies, based on the constraints and goals they dictate and contextual information;
- **Monitoring, Auditing and Reporting**: this step is about monitoring the enforced policies, logging relevant information and auditing the environment, in order to check for violations and report anomalies or unexpected/unplanned situations.

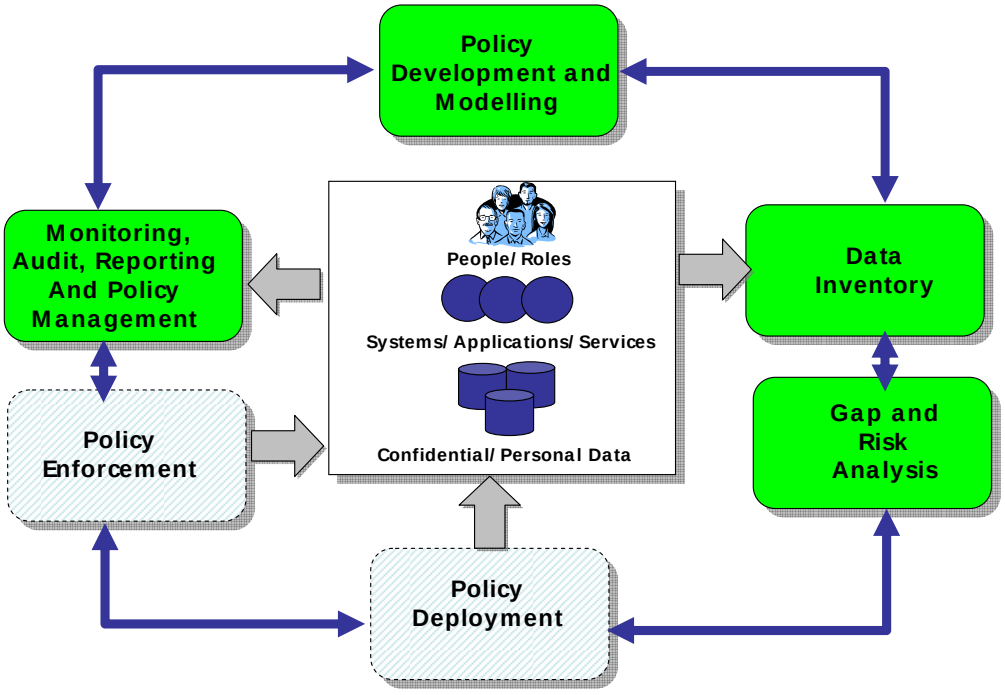


Figure 10: Data Governance and Policy Management Process

This process is dynamic as policies and data are subject to periodic changes driven by business, security, trust or privacy needs. Multiple iterations and refinements could occur.

In terms of privacy policy management for personal data, most of the current work done (and the available solutions) is in the areas of policy developing and modelling, data inventory, risk and gap analysis, monitoring, auditing and reporting. These areas have been represented with the “green” colour in Figure 10.

In general privacy policies can be hard to enforce via IT solutions. The enforcement of privacy rights, permissions and obligations related to confidential and personal data requires the mapping of these concepts (that are most of the time abstract and based on high-level princi-

ples) into rules, constraints and access control, the meaning of which must be unambiguous so that it can be deployed and enforced by software solutions.

Dealing with this still requires that the entities involved in the management of confidential and personal data follow best practices and good behaviours. However, being able to automate aspects of the enforcement of privacy policies and reduce the involved costs is important for enterprises.

Advancements in this direction have already been made when dealing with the (technological) enforcement of privacy rights and permissions. Extended access control and authorization mechanisms have been built to check privacy permissions against users' rights, the stated purposes of the confidential information (that needs to be accessed) and the declared intents [CaTB05]. This is the case, for example, of web transactions and interactions or applications/services within organizations that need to access and manipulate confidential data for business reasons.

More complex is the case of dealing with privacy obligations. Privacy obligations define and describe the expected behaviours and constraints to be satisfied by enterprises when handling confidential and personal data. They dictate a privacy-aware lifecycle management of personal and confidential data.

They might include the deletion of confidential data after a predefined (potentially very long) period of time, periodic notifications and request for authorization to data owners or data subjects, dealing with consent before executing actions (e.g. data transfer outside the enterprise or across borders), fulfilment of opt-in/opt-out choices made by data owners, data transformation and minimization, ongoing compliance with laws' obligations and internal guidelines.

The events that trigger the fulfilment of privacy obligations can be completely orthogonal to the ones relevant for access control. For example, the obligation dictating the deletion of personal data after 7 years has to be fulfilled independently if these data have ever been accessed.

Privacy obligations can have ongoing aspects that need to be monitored and satisfied over a long period of time. These tasks are challenging for enterprises because of the need for specific IT infrastructures and processes able to manipulate confidential data as dictated by privacy obligations.

Enterprises have been investing on identity management solutions in the last few years. Privacy management solutions, and in particular solutions dealing with privacy obligations, need to leverage and integrate with these investments in order to be successfully adopted.

In general, the management and enforcement of privacy obligations, as first class citizens (i.e. without being considered just as a secondary and subordinated aspect of access control), is still a green field and open to research.

Next chapters better qualify the addressed problem, analyse some of the related issues, describe a technical approach to move towards a more explicit management and enforcement of privacy obligations and introduce a trusted system dealing with these tasks.

3 Addressed Problem: Management and Enforcement of Privacy Obligations

This thesis addresses the problem of dealing with the explicit and automated management of privacy obligations in enterprises.

Related questions and issues that need to be properly analysed include:

- What are the core aspects of privacy obligations that must be handled?
- How can we represent privacy obligations?
- How can we associate privacy obligations to the personal data they refer to?
- How to manage them?
- What are the core management functions that need to be put in place when handling privacy obligations?
- How to enforce them? How to do this in the context of current IDM systems?
- How to address related regulatory compliance issues?

The main part of our work focuses on the analysis of privacy obligations and the definition of mechanisms and solutions to deal with the representation/modelling, enforcement and monitoring of privacy obligations.

We also explore the related problems of managing the strong association of privacy obligations to data, enforce accountability and provide more transparency to users.

We believe that a reliable and verifiable management of personal data, in accordance with legal requirements and the policies of the data subjects, can be more easily achieved if it is controlled by privacy specific middleware rather than by application-level code. After all, the driving force behind any application solution is the set of business processes for which it is designed, not the privacy management aspects of the personal data it processes. The use of privacy management middleware allows a common, systematic (as opposed to piecemeal) approach to privacy issues to be taken, thereby creating trusted systems.

This approach has been broadly experimented by identity management solutions and validated by successful deployments and usages of these solutions.

By following this principle, we are also looking at middleware approaches to handle privacy obligations.

In this context, the management of privacy obligations is a green field. Work has already been done to address some of the issues, in particular related to the representation of privacy policies (and obligations), their enforcement in transactional and interaction-driven contexts and the management of simple long-term aspects of obligations for data retention. In many cases, though, obligation policies are considered as second-class entities the enforcement of which is subordinated to other aspects of privacy policies, such as access control. A more explicit and comprehensive approach to privacy obligations is required.

We aim at researching and building a system where privacy obligations are considered as first-class “citizens” that can be managed without their subordination to other aspects. Our goal is to ensure that this work can be deployed into current state-of-the-art enterprise identity management solutions to allow enterprises to leverage their current investments in this area.

4 Analysis of Privacy Obligations: Common Aspects and Requirements

Privacy obligations define and describe the expected behaviours and constraints to be satisfied by enterprises when handling confidential and personal data. They dictate a privacy-aware data lifecycle management including data retention and deletion aspects, management of notifications and requests for authorization, data processing and transformation workflows.

Enterprises need to put in place underlying IT infrastructures, processes and mechanisms to be compliant with these obligations. This can be a challenging task due to the fact that privacy obligations can differ quite substantially given their current level of refinement (abstract vs. refined) and their "multidimensional" nature involving multiple factors and aspects.

4.1 Abstract vs. Refined Privacy Obligations

Privacy obligations can be very abstract and generic, for example: ***"every financial institution has an affirmative and continuing obligation to respect customer privacy and protect the security and confidentiality of customer information"*** - Gramm-Leach-Bliley Act [GLB03].

This type of obligations dictates high level principles and guidelines that need to be interpreted, refined and grounded to specific contexts in order to be fully understood in terms of their operational implications.

More refined privacy obligations can be expressed in terms of:

- **notice requirements**;
- **opt-in/opt-out options limitations on reuse of information and information sharing for marketing purposes**;
- **data retention and deletion limitations**.

At the other extreme, privacy obligations can dictate very specific requirements.

This is the case where data retention has to be enforced for a long period of time or data are temporarily stored by organisations: privacy obligations can require that personal data must be deleted after a predefined number of years, e.g. 30 years (i.e. long-term commitment) - or in a few days if user's consent is not granted (i.e. short-term commitment).

Other very specific privacy obligations might require the enterprise to notify (for example via e-mail) the data subjects, in case their data has been accessed by third parties or unauthorised people (for example in case of hacking or identity frauds).

Similarly, privacy obligations might mandate to execute well defined workflows and processes, involving both humans (e.g. for explicit request for authorization) and computer systems in presence of specific events.

4.2 Multidimensional Nature of Privacy Obligations

Privacy obligations depend on and are influenced by a variety of aspects, including data subjects' preferences, enterprise guidelines, legislation and, once refined, technical aspects.

Figure 11 is an attempt to capture this multidimensional nature of privacy obligations, based on our current analysis of privacy obligations [Casa04a], [Casa04b] and their implications in terms of life-cycle management of personal data.

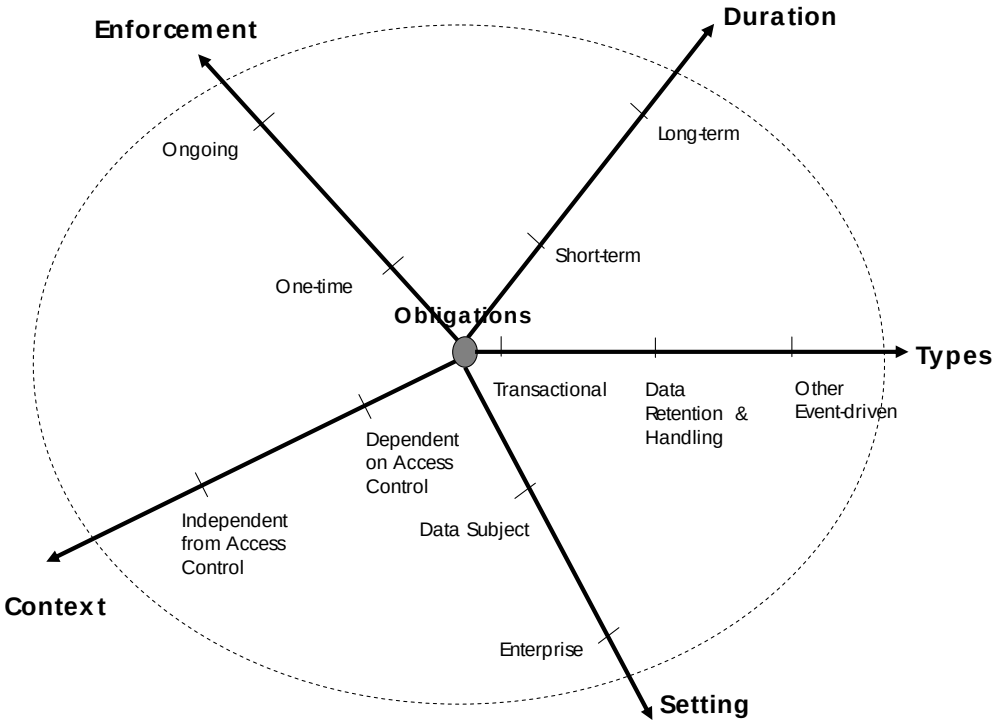


Figure 11: A multi-dimensional View of Privacy Obligations

The key dimensions/aspects that need to be considered to characterise privacy obligations are:

- **Types of obligations**: obligations can be classified based on the fact that they are:
  - **Transactional**: these obligations need to be fulfilled immediately, during a transaction or interaction, when accesses to personal data are required;
  - **Data Retention & Handling**: these obligations related to the management of personal data in terms of their deletion or transformation. They can be long-termed and not really related to accesses to data;
  - **Other Event-driven obligations**: these obligations are triggered by events that can be dictated by contextual and system information, such as location of systems, their trustworthiness, aggregated meta-information associated to data (such a access counters, etc.);

- **Duration**: obligations can be classified based on their "lifetime" i.e. the period of time where they are active and subject to enforcement:
  - **Short-termed**: privacy obligations could be short-termed. This is the case of transactional obligations or obligations the lifetime of which ranges in the order of few hours to a few months;

  - **Long-termed**; privacy obligations could be long-termed. This applies to all cases where data retention period could span to the order of years and consequently obligations need to be fulfilled over that period of time;

- **Enforcement**: obligations can be classified based on their enforcement implications:
  - **One-time**: this is the case where a privacy obligation can be considered as being fulfilled once it has been enforced. For example, an obligation dictating the deletion of a piece of data at a specified point of time belongs to this category;
  - **Ongoing**: this is the case where a privacy obligation might require to be "enforced" multiple times, during its lifetime. For example, this is the case of obligations dictating periodic notifications, over a predefined period of time;

- **Context**: obligations can be classified based on the context where they operate and are likely to be triggered for fulfilment:
  - **Access control context**: privacy obligations can be triggered as an effect of accessing data. This is the case, for example, of transactional obligations;
  - **Access control-independent context**: privacy obligations can be triggered in context completely independent from access control, for example deletion of data at a due period of time;

- **Setting:** obligations can be set by different entities:
  - **Data subjects:** data subjects could define privacy obligations to be fulfilled on their data, for example by specifying opt-in, opt-out options that are transformed into obligations for enterprises. This can include deletion and notification preferences. Alternatively, trusted third parties, acting on behalf of data subjects, could do this, for example identity providers in federated identity management contexts;
  - **Enterprise**: administrators within the enterprise might define privacy obligations on data, as dictated by internal guidelines and/or legislation.

Figure 12 shows two simple examples of privacy obligations and their mapping in this multi-dimensional space.

The first example of privacy obligation, ***"Notify UserA via e-mail if his/her Data is Accessed",*** dictates data handling criteria. It can be set by the data subject on his/her account (for the entire lifetime of this account). It requires multiple enforcements (every time personal data is accessed). This obligation is triggered by accesses to the personal data.

The second example of privacy obligation, ***"Delete Data XYZ after 7 years"***, can be set by an enterprise privacy administrator. It has long-term implications but it requires one-time enforcement (deletion of data at a predefined period of time). It is independent from access control aspects: data has to be deleted independently from the fact if it has ever been accessed.

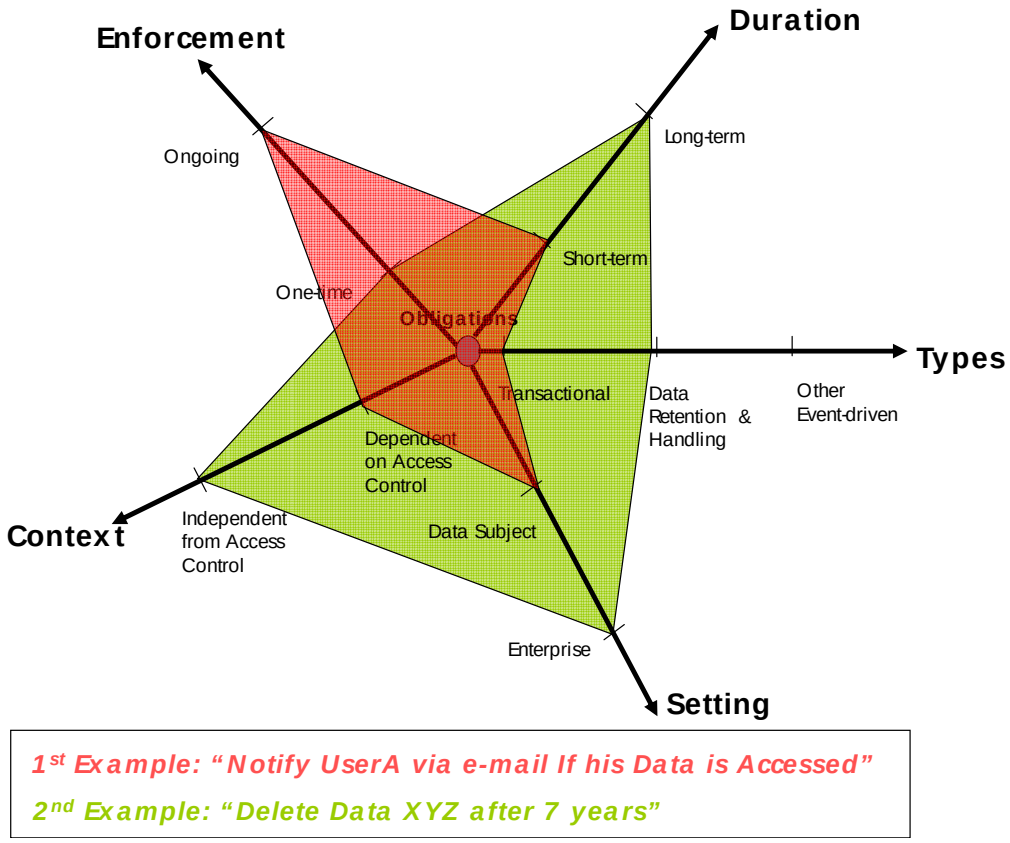


Figure 12: Simple Examples of Privacy Obligations

4.3 Common Properties and Aspects of Privacy Obligations

Our analysis of privacy obligations [Casa04a], [Casa04b] is based on current privacy laws, privacy guidelines and customers' requirements. It has identified a set of core properties that are shared by privacy obligations:

1. **Period of validity of an obligation:** it is the lifetime of an obligation i.e. the period of time where the obligation is "active" and needs to be managed (enforced and monitored);
2. **Degree of enforceability of an obligation:** the enforcement of privacy obligations can be automated or, in some cases, it might need to involve human processes and best practices;
3. **Target (involved data) of an obligation:** privacy obligations refer to personal data subject to these obligations. Different, heterogeneous types of data, stored in multiple data repositories, can be referenced by a privacy obligation;
4. **Events that trigger the need to fulfil an obligation:** privacy obligations can be triggered by one or more events (for example time-based events). Logical combi-

nations of events (involving AND, OR and NOT operators) might be required to express the conditions under which privacy obligations need to be enforced;

5. **Actions that need to be executed to enforce an obligation:** the enforcement of an obligation might require the execution of one or more actions. These actions could be as simple as deleting data or notifying people or require the execution of complex workflow that might involve human and computer interactions**;**
6. **Entities that are responsible for enforcing an obligation:** for each obligation it should be clear who (organisation, group, individual) is responsible for their management and enforcement**;**
7. **Accountability criteria:** these criteria mainly define logging and auditing requirements, to ensure that the system keeps an historical track of how an obligation is managed and enforced and which violations occurred;
8. **Exceptions:** exceptional cases might need to be analysed and explicitly described, in order to assure a correct management and enforcement of obligations.

Part of these privacy obligations can be enforced by software systems i.e. tools can be built in order to manage and automate their fulfilment, based on the expressed constraints and requirements. Other privacy obligations, dictating expected human behaviours still need to rely on best practices and good behaviours of enterprises and employees. Nevertheless, we believe that the process of moving towards automation (for those obligations where this is possible) is useful to enterprises to help them in their governance, regulatory compliance and cost reduction efforts.

In our work we focus on automatically enforceable privacy obligations. Concepts and approaches described in the remaining part of this thesis can still apply to other types of privacy obligations, at least with regard to the modelling aspect and the analysis of related requirements.

We specifically focus on the requirements and issues related to the management and enforcement of the following three core categories of privacy obligations:

1. **Long-term privacy obligations;**
2. **Short-term and transactional privacy obligations;**
3. **Ongoing privacy obligations.**

Table 1 shows a few examples of events and actions related to these types of privacy obligations.

More comprehensive and complete examples of privacy obligations are described in Chapter 6 and Chapter 7.

Specifically, chapter 6 describes our model of privacy obligations and analyses it from different perspectives: privacy obligation examples are provided from an operational perspective.

Chapter 7 describes the actual (XML-based) format used to represent privacy obligations in the prototype developed at HP Labs.

<table>
<tr><th colspan="4">Long-term Privacy Obligations</th></tr>
<tr><th colspan="2">Events Triggering Obligations</th><th colspan="2">Actions Dictated by Obligations</th></tr>
<tr><td>Time-driven</td><td rowspan="2">1. at a specific date and time (e.g. 1:00am 01-Jan-2005)
2. after a certain period of time (e.g. 1 hour, 3 days, 5 minutes)
3. after the data has being used for a certain number of times (e.g. after being used twice) in a specific time-frame</td><td>Delete/ Update</td><td>1. delete all confidential data of a given data subject
2. partially delete data (e.g. delete only the credit card number)
3. replace data with an up-dated set of data (e.g. up-date subject's address)</td></tr>
<tr><td>Driven by Us-age and Count-ers</td><td>Hide/ Unhide</td><td>1. hide (encrypt) all data of a subject from any access
2. hide a part of this data from any access
3. unhide all data
4. unhide a part of the data</td></tr>
<tr><th colspan="4">Ongoing Privacy Obligations</th></tr>
<tr><th colspan="2">Events Triggering Obligations</th><th colspan="2">Actions Dictated by Obligations</th></tr>
<tr><td>Time-driven</td><td>1. periodically (e.g. every month)</td><td></td><td>1. send a report to a subject containing the status of their data and their opt-in/opt-out options (e.g. number of times being used, who has tried to ac-cess)
2. tell the subject what data he/she has provided
3. get updated data from sub-ject
4. audit the logs, report any improper use of the data</td></tr>
<tr><td rowspan="4">Driven by Con-textual Events</td><td rowspan="4">1. when the data being used
2. when the data being transferred
3. when the data being deleted
4. a particular party/parties try to access
5. data is being used for certain purpose (e.g. send advertisement)
6. a set of data is going to be retrieved together
7. any action predefined by the data subject</td><td>Notify</td><td>1. notify the subject</td></tr>
<tr><td>Log</td><td>1. take logs</td></tr>
<tr><td>Access</td><td>1. default allow/disallow all access
2. allow
3. disallow</td></tr>
<tr><td>Consult</td><td>1. get authorization from data subject
2. get authorization from third party
3. check according to certain condition made by the user</td></tr>
<tr><td>Others</td><td>1. when the privacy policies changed</td><td></td><td>1. Stop access to the data
2. update obligation</td></tr>
<tr><th colspan="4">Short-term and Transactional Privacy Obligations</th></tr>
<tr><td colspan="4">Obligations might need to be dictated by a transaction or an interaction. The actions specified by these obli-gations might need to be immediately fulfilled. These actions can be the same as the ones specified by long-term and on-going obligations.</td></tr>
</table>

Table 1: Types of privacy obligations and examples of related events and actions

4.4 Important Issues and Requirements

To categorize core issues and requirements related to the management and enforcement of privacy obligations we analysed a few scenarios involving the management of digital identities and identified a few common patterns:

- **Enterprise scenario**: personal data are collected from customers, employees and business partners. They are accessed, used and processed to enable business transactions and processes. Data can be disclosed to business partners and/or third parties;
- **E-commerce scenario**: personal data are collected from customers, mainly to enable business transactions and for marketing purposes. Data can be disclosed to third parties;
- **Healthcare scenario**: medical and personal data are collected from patients. Data can be accessed by medical people and shared with third parties for research and medical reasons;
- **Government scenario**: personal and financial data are collected from citizens by government offices (Revenue Office, Pension Office, Home Security Office, etc.) to provide government services and for security reasons;
- **Federated identity management scenario**: this scenario is complementary and orthogonal to the above scenarios. It is about dealing with explicit federated environments, where personal data and identities are shared among multiple parties (usually within a circle of trust or based on contractual agreements) to enable single-sign-on and speed-up the authentication process.

In these scenarios data subjects (people) directly or indirectly disclose their personal data to enterprises (organisations). In doing this they might be asked (or want) to specify their privacy preferences, for example in terms of opt-in/opt-out choices, requests for notifications, retention, usage and disclosure of their data for predefined purposes.

Enterprises using modern identity management solutions can provide self-registration and user provisioning tools that allow users to retain control of part of their data and specify (and change overtime) some of their requirements and preferences. Some of these preferences must be translated into explicit privacy obligations to allow for their automated management within organisations, such as obligations to notify data subjects about usages of their data, delete data, protect data, etc. A few important questions arise.

How can privacy preferences be translated into privacy obligations?

Which format should be used to represent privacy obligations?

How are links and associations between privacy obligations and stored data going to be handled?

Privacy administrators within these enterprises might need to set up additional privacy obligations on stored data, to fulfil privacy laws and/or internal guidelines. This might apply to all information involving personal data, including data subjects' records, audit logs, documents, etc.

Which tools are required by administrators to manage and check these obligations on a large database containing personal data?

How would these tools fit in current identity management solutions?

How to ensure that enterprises will handle these data and related obligations in an accountable way?

In all these scenarios, personal data might be exchanged across boundaries, e.g. with other organisations, to enable interactions, transactions or business processes. If these data are subject to privacy obligations, obligations need to be communicated as well. In some cases they must be modified and adapted, depending on the location and nature of the data receivers.

How to ensure that privacy obligations are "strongly" associated to these data and will be enforced?

Our investigation identified the following important issues and requirements which need to be considered when dealing with the management and enforcement of privacy obligations:

1. **Explicit modeling of privacy obligations**: to be managed, privacy obligations need to be represented with an appropriate language to describe which data is affected by an obligation, the events and conditions that trigger the fulfilment of the obligation, actions to be carried on, which entities are responsible and accountable for their enforcement;
2. **Association of obligations to data**: the association of privacy obligations to the targeted confidential data must not be easy to be broken. This aspect is particularly challenging in dynamic environments where confidential data can be moved around or sent to other parties;
3. **Mapping obligations into actions**: when possible, actions and sequences of actions dictated by obligations must be expressed in a way that can be programmatically enforced; otherwise, they should trigger related processes and workflows involving the human intervention and clearly stated responsibilities;
4. **Compliance of refined obligations to high-level policies**: the mapping of high level policies to refined privacy obligations (and the affected data) should be managed explicitly and tools built to spot potential inconsistencies and dependencies;
5. **Tracking the evolutions of obligation policies**: obligation policies can be carried on over long periods of time and are subject to changes. Changes need to be tracked and obligations versioned, for accountability reasons and to deal with the evolution of the contexts and frameworks where these obligations apply;
6. **Dealing with long-term obligation aspects**: long-term obligations have implications on the longevity and survivability of related processes and the involved data. Solutions need to be build to last over a long period of time;
7. **Accountability management**: as anticipated before, accountability management is fundamental to ensure that the enforcement of privacy obligations is carried on with clear responsibilities of the involved parties. This introduces requirements in terms of auditing, tracking of obligations and their monitoring;
8. **Monitoring obligations**: the fulfilment of obligations must be monitored and checked against expected situations and behaviours. Despite good intents and enforcement mechanisms, it can always happen that the fulfilment of obligations is omitted. Monitoring mechanisms must be orthogonal to the enforcement mechanisms. Problems need to be notified to the responsible entities;

9. **User involvement and awareness**: users should have visibility of which obligations an organisation has with them. Tools should be provided to uses to allow them to monitor their fulfilment and directly manage their privacy obligations;
10. **Complexity and cost of instrumenting applications and services**: the enforcement and monitoring of obligation policies can have an impact on the involved applications and services, both in terms of their instrumentation and development costs. A privacy obligation framework should reduce to the minimum this impact.
11. **Integration with current identity management solutions**: systems that manage and enforce privacy obligations must integrate with current state-of-the-art identity management solutions.

5 Related Work

The management and enforcement of privacy obligations can be a reasonably easy task when the events that trigger them are well defined and simple to capture, for example they depend on time or known transactions or interactions. More complex is the case of privacy obligations related to ongoing obligations, triggered by the occurrence of events and conditions non-necessarily related to time or known transactions (for example dictated by laws, user's requests, etc.).

Relevant work has been done by W3C with their Platform for Privacy Preferences Project (P3P) specification [W3C02] to allow people to describe in more details their privacy expectations/preferences and match them against the level of privacy supported by an enterprise. Based on [W3C02]: "The Platform for Privacy Preferences Project (P3P), developed by the World Wide Web Consortium (W3C), is emerging as an industry standard providing a simple, automated way for users to gain more control over the use of personal information on Web sites they visit. At its most basic level, P3P is a standardized set of multiple-choice questions, covering all the major aspects of a Web site's privacy policies. Taken together, they present a clear snapshot of how a site handles personal information about its users. P3P-enabled Web sites make this information available in a standard, machine-readable format. P3P enabled browsers can 'read' this snapshot automatically and compare it to the consumer's own set of privacy preferences. P3P enhances user control by putting privacy policies where users can find them, in a form users can understand, and, most importantly, enables users to act on what they see".

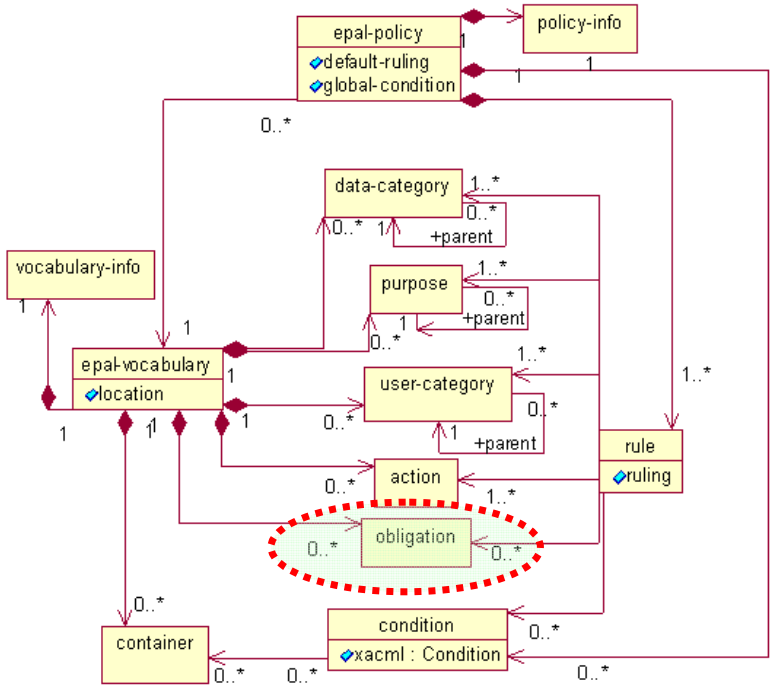
P3P is important to shape (aspects of) the trust that people might have on the enterprise by verifying which privacy aspects they promise they can fulfill. However P3P is mainly a "front-end" mechanism, in the context of web services. In its current form it is "passive" i.e. it only checks if people's expectations are matched against promises made by the enterprise. It does not address the problem of allowing users to express fine grained privacy obligations; it does not provide mechanisms to deal with the execution and fulfillment of these privacy obligations and related constraints by enterprises. Last but not least, it does not define an enterprise framework for dealing with privacy policies.

Relevant work in the space of privacy management for enterprises is described in [KaSc02], [KaSW02a], [ScAs02], [KaSW02b]. An Enterprise Privacy Architecture (EPA) is introduced and described in [KaSW02b], encompassing a policy management system, a privacy enforcement system and an audit console.

Specifically, [ScAs02] introduces additional architectural details about EPA along with an interpretation of the concept of privacy obligations. This concept is framed in the context of privacy rules (policies) defined for authorization purposes.

This approach is further refined and described in the Enterprise Privacy Authorization Language (EPAL) specification [Epal04], currently submitted to W3C for standardization.

Figure 13 shows the UML schema of a privacy policy, as defined in EPAL:



Source: http:/ / www.w3.org/ Submission/ 2003/ SUBM-EPAL-20031110/

Figure 13: EPAL privacy policy – UML Diagram

The current EPAL specification does not provide a format (or description) of obligations: obligations are purely a placeholder in the policy rule.

In EPAL the privacy management framework is also framed around the concept of access control and, conceptually, it looks like the one shown in Figure 14.

Attempts of users, applications and services to access personal and confidential information is intercepted and mediated by an access control system, driven by EPAL policies. The enforcement of associated obligations can be triggered during an access control decisions.

Hence in EPAL privacy obligations are seen as entities subordinated to access control: this is not necessarily correct or complete, as privacy obligations might be totally independent from access control aspects. For example, the deletion of data in 7 years' time has to happen independently if these data has ever been accessed.

Similarly to EPAL, XACML by OASIS [OASI05a] specifies the syntax and format of access control policies and related obligations. The approach is the same, i.e. privacy obligations are subordinated to access control policies.

A recent research article [Ande05] compares EPAL vs. XACML. Among other things, this article draws the following conclusions: "in almost every area, the functionality of XACML 2.0 is a superset of EPAL 1.2. Where the two languages differ, the EPAL differences often result in less functionality than XACML has. In many cases, the EPAL 1.2 differences from XACML make construction of flexible privacy policies impossible or difficult".

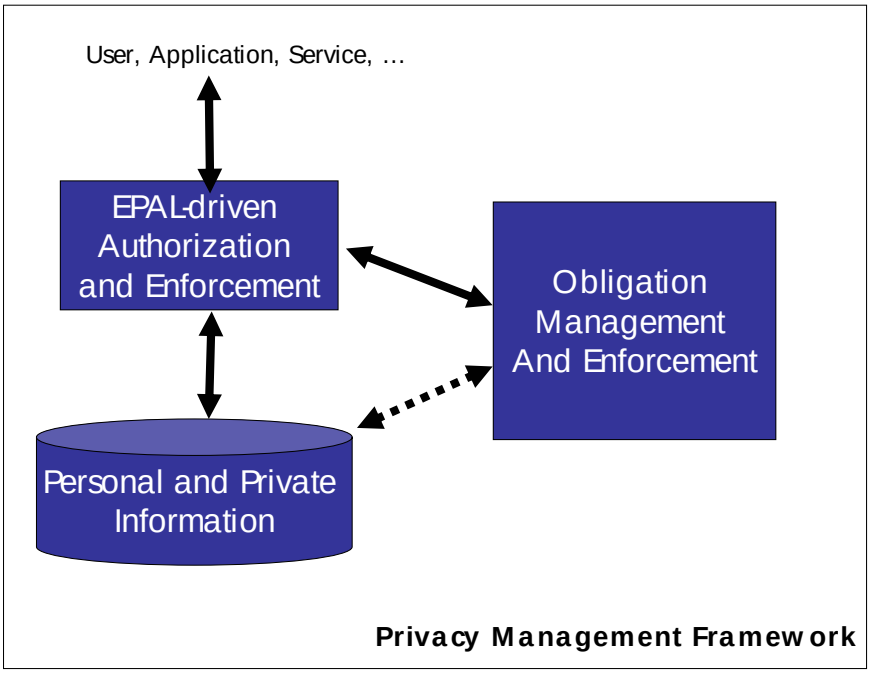


Figure 14: EPAL - Privacy Management Framework

The above work makes important advancements in exploring and addressing the problem of privacy management in enterprises but it only considers the authorization and access control perspective as the driver for their representation, management and enforcement.

It has still to be fully demonstrated that privacy obligations can be managed at their best from an authorization-based perspective. Privacy obligations can include aspects that are not really driven by authorization aspects, such as dealing with the deletion of confidential data at a specific date/event, periodically providing notifications to subjects about stored confidential data, triggering workflows based on contextual changes, dealing with ongoing requests dictated by subjects or laws.

We believe that the representation, management and enforcement of privacy rights, obligations and permissions should be addressed without imposing any specific or dominant perspective.

In our proposed approach (described in the next chapters) obligation policies are “first-class citizens” that are explicitly managed. Even if our architecture has high-level commonalities with the architecture described in [KaSc02], [KaSW02a], [ScAs02], [KaSW02b] we further refine the concept of obligations and we introduce the concept of obligation versioning and tracking.

We also split the enforcement mechanisms in two parts by including a scheduling mechanisms and an obligation enforcer where the obligations actions are carried out by flexible workflow processes that allows both automation and the involvement of people.

Mechanisms to deal with (privacy) obligations have already been implemented in products, in particular for data retention, for example [Ibmt04] and in a variety of document and record

management systems. Nevertheless, these approaches are very specific; they are focused on the particular domain of record and document management and handle simple obligation policies (such as deletion or retention of data).

They are expensive solutions that need to be integrated with enterprise document management systems. They are not really designed to be deployed in operational identity management system that handle day-by-day activities on personal data, identity information and user accounts.

Our work aims at pushing the barrier even further to create an obligation management framework that can be leveraged by and integrated with modern, state-of-the-art identity management solutions, for a variety of purposes and tasks (including privacy-aware management of user provisioning, audit logs, etc.).

Work has been done to represent privacy policies, including obligations such as [Epal04], [BJSW02], [DDLS01]. Some of the core privacy concepts have been leveraged by these papers though the lack of specification of what privacy obligations are and a suitable format has required further research and specification from our side.

From a compliance management perspective, it is important to monitor privacy obligations and check for violations. This is not a new concept and it is at the base of current monitoring and auditing systems that, more in general, check for policy compliance against events and audit logs. Specific work describing the monitoring of obligations (seen as an aspect of access control), in the context of policy management, is described in [DDLS01]. Our work also provides basic monitoring mechanism to check for violations of enforced privacy obligations.

Relevant work on mechanisms to associate policies to data is described in [KaSc02], [KaSW02a], [ScAs02], [KaSW02b], [CaPB03], [AKSX02]. Each mechanism has pros and cons in terms of the implications for existing enterprise applications, services and data repositories.

We can leverage aspects of this work, in particular [CaPB03] to provide a stronger association of obligation policies to confidential data by using cryptographic mechanisms to encrypt data along with the associated policies and check for compliance by using one or more trusted third party’s services.

However, more research has to be done in this space to verify the actual usability and scalability of the proposed approach.

6 Our Model of Privacy Obligations and Related Management Framework

Based on the available requirements and the analysis of the limitations of current solutions, we introduce and describe an alternative privacy obligation management model where privacy obligations are considered as “first class” entities and introduce an explicit privacy obligation management framework to handle these obligations.

The details about our model and related concepts follow.

6.1 Model of Privacy Obligation Framework

An obligation management framework is introduced to explicitly handle privacy obligations. Figure 15 shows the conceptual model underpinning this framework.

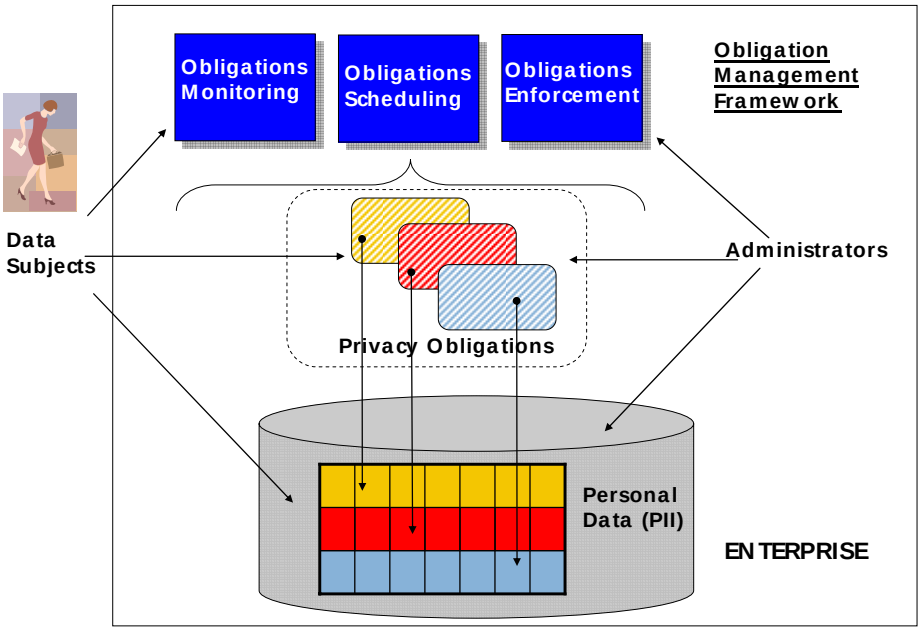


Figure 15: Proposed Privacy Obligation Management Model

In our model privacy obligations are independent entities that are explicitly modeled and managed to enable a privacy-aware lifecycle management of personal data. They are not subordinated to access control aspects.

Data subjects can define privacy obligations and associate them to their personal data at the disclosure time (e.g. during a self-registration process) or at any subsequent time.

Enterprise privacy administrators can also associate additional privacy obligations, for example dictated by laws or internal guidelines.

In our model, the obligation management framework handles these obligations and their associations to personal data by providing the following core functionalities:

- **Explicit modeling and representation of privacy obligations:** a language/format is defined to explicitly represent privacy obligations in order to analyse them and reason about their implications;
- **Scheduling the enforcement of privacy obligations**: the system schedules which obligations need to be fulfilled and under which circumstances (events);
- **Enforcing privacy obligations**: the system enforces privacy obligations once they are triggered. The enforcement ranges from the execution of simple actions to complex workflow involving human interventions;
- **Monitoring the fulfilment of privacy obligations**: the system monitors and audits the enforced obligations, at least for a predefined period of time, to ensure that the desired status of data is not violated and to report anomalies;
- **Administration and lifecycle management of privacy obligations**.

These functionalities can be accessed by enterprise privacy administrators and potentially by data subjects, for example to monitor their personal data and check for privacy compliance. Additional details about our model can be found in [Casa04a, Casa04b]. Chapter 7 describes how this privacy obligation framework has been implemented for real, by an obligation management system. Chapter 9 describes how this obligation management system can be integrated with an identity management solution.

6.2 Model of Privacy Obligations

Our model of privacy obligations can be analysed by means of different (but equivalent) views/perspectives:

- **Conceptual view;**
- **Formal view**;
- **Operational view**.

6.2.1 Conceptual View

From a **conceptual perspective**, a privacy obligation can be considered as an entity (object) with a few associated properties, as shown in Figure 16.

In this view, a privacy obligation is characterised by the following core properties:

- **Obligation Identifier**: it is an identifier to uniquely identify an obligation within the entire obligation management system;
- **Targeted Personal Data**: it is a list of references to personal data that are affected by this privacy obligation. A reference must include all the information necessary to reach the data, though it can be codified in a way to avoid any indirect exposure (or correlation) of personal data.
- **Triggering Events**: it is a list of logical (AND/OR) expressions based on combinations of basic events (e.g. time, access, counters) that can trigger the need to enforce the privacy obligation;

- **Actions**: it is a list of actions to be executed at the enforcement time of the privacy obligation. Actions could be very simple - such as deletion of data or sending a notification - or much more complex, for example workflow involving both system and human interaction steps.
- **Additional Metadata**: it is a placeholder for additional properties still under exploration, such as exceptions, accountability constraints, versioning and integrity check, etc.

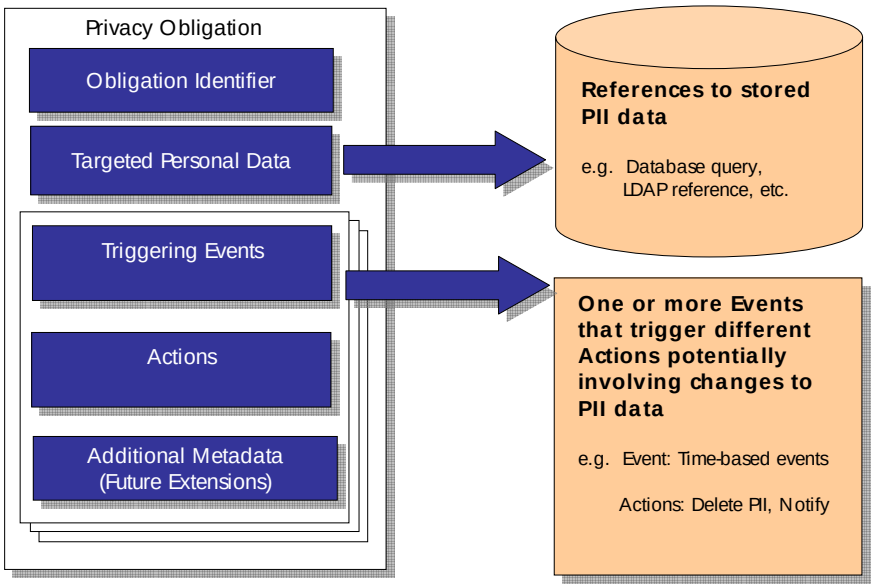


Figure 16: Model of a Privacy Obligation

6.2.2 Formal View

From a **formal perspective**, a privacy obligation can be seen as a **<*i,t,L(e),C(a)*>** tuple, where **<*i,t,e,a*>** ∈ **<I, 2^T, 2^E, 2^A >**:

- **I**: set of unique identifiers, associated to obligations;
- **T**: set of possible obligation targets, i.e. data entities (e.g. personal data, digital identities, attributes, etc.) subject to obligations;
- **E**: set of possible events that can trigger an obligation;
- **A**: set of all possible actions that can be executed as an effect of enforcing an obligation.

Specifically, a **<*i,t,e,a*>** tuple is defined as follow:

i ∈ **I**: ***i*** is an *element* that belongs to **I**;

t ⊆ **T**: ***t*** is a *set* of targets included in **T**;

e ⊆ **E**: ***e*** is a *set* of events included in **E**;

a ⊆ **A**: ***a*** is a *set* of actions included in **A**.

A privacy obligation is obtained by applying the ***L*** operator to the ***e*** set and the ***C*** operator to the ***a*** set:

- ***L(e)***: defines a logical combination of events, for example AND, OR and NOT combination of events contained in *e*;
- ***C(a)***: defines an operational combination of actions, such as a sequence of actions.

It is beyond the scope of this thesis to provide a systematic definition or formalization of privacy obligations.

In this thesis we will have a pragmatic view of privacy obligations, based on how we can represent them and how we can operate on them.

6.2.3 Operational View

From an **operational perspective**, privacy obligations can be seen as *reactive rules* [RGC+05] i.e. rules that are triggered by events and/or by the fact that the specified conditions are met. As an effect (reaction) of triggering a rule, actions are executed.

A representation of privacy obligations as **reactive rules** follows:

OBLIGATION Oid:
TARGETS: t
WHEN L(e)
EXECUTE C(a)

In this context, given an obligation with unique identifier *Oid* and a target *t*, if the logical combination of events *L(e)* is true, i.e. it triggers the rule, then the combination of actions *C(a)* has to be executed.

The remaining part of this thesis will focus on this operational definition of privacy obligations.

6.2.3.1 Examples of Privacy Obligations

As anticipated at the beginning of this chapter, privacy obligations are associated to personal data and can be defined by data subjects and privacy administrators.

A few simple examples of privacy obligations follow:

1) ***OBLIGATION Oid1:***
TARGETS:
t1:< DATABASE=db1, TABLE=customers, Key=CustomerName, KeyValue=abc>
WHEN (current_time= date1)
EXECUTE <DELETE t1>

In this example, a customer record, stored in a specified table of a database, must be deleted at a well defined point of time. This is a simple example of a data deletion obligation.

```
2) OBLIGATION Oid2:
TARGETS:
  t1:< DATABASE=db1, TABLE=customers, Key=CustomerName, KeyValue=abc,
  ATTRIBUTES=(e-mail) >
WHEN (Access_Data_Event AND Access_Data_Event.data = t1)
EXECUTE  <NOTIFY BY t1.e-mail>
```

In this example, when an event (for example issued by an access control system) indicates that a specific customer's record has been accessed, a notification has to be sent to the customer, by using his/her e-mail address.

```
3) OBLIGATION Oid3:
TARGETS:
 t1:< DATABASE=db1, TABLE=customers, Key=CustomerName, KeyValue=abc
ATTRIBUTES=(creditcard,e-mail)>
WHEN
   (current_time>date1)
     AND
  (NOT (Access_Data_Event AND Access_Data_Event.data = t1 ))
EXECUTE
  <NOTIFY BY t1.e-mail>
  <DELETE t1.creditcard>
```

In this example, if customer's data is not accessed after a predefined amount of time, an attribute (credit card) has to be deleted and the customer must be notified.

```
4) OBLIGATION Oid4:
TARGETS:
  t1:< DATABASE=db1, TABLE=customers, Key=CustomerName, KeyValue=abc
  ATTRIBUTES=(creditcard,e-mail)>
WHEN
   (current_time>date1)
     OR
     ( (Access_Data_Event AND Access_Data_Event.data = t1 )
       AND
      (Access_Counter>n))
EXECUTE
```

```
<DELETE t1.creditcard>
<RUN WORKFLOW deprovision_user(t1.KeyValue)>
```

In this example customer's data is deleted and the customer account is de-provisioned (accounts deleted, access rights revoked, etc.) from various IT systems either at a specific point in time or after customer's data has been accessed more than *n* times.

```
5) OBLIGATION Oid5:
TARGETS:
 t1:< DATABASE=db1, TABLE=customers, Key=CustomerName, KeyValue=abc,
 ATTRIBUTES=(e-mail)>
WHEN
 (current_time < date1)
  AND
 (time_counter > time_interval)
EXECUTE
 <NOTIFY BY t1.email>
 <RESET time_counter>
```

In this example, periodic (ongoing) notifications are sent by e-mails to customers, for example to notify them about the fact that the enterprise is retaining their personal data. This is an example of ongoing obligation.

Specific types of privacy obligations can be set-up by enterprise privacy administrators to handle personal data based on internal guidelines and/or laws. These privacy obligations can be triggered by internal events determined by contextual and infrastructural changes. A few examples follow.

```
6) OBLIGATION Oid6:
TARGETS:
 t1:< DATABASE=db1, TABLE=customers>
WHEN
   (Event-intrusion_detected)
EXECUTE
  <ENCRYPT  t1>
  <NOTIFY admin>
```

In this example, we assume that an intrusion detection system is able to send alerts to subscribers (including our obligation management system) when intrusion attempts are detected. A privacy obligation can be triggered to protect the entire content of a "confidential" table by

encrypting its content and notifying the administrator. This action can be seen as a best-effort, temporary solution to prevent that personal data are accessed by the intruder.

7) OBLIGATION Oid7:
TARGETS:
t1:< DATABASE=db1, TABLE=customers>
WHEN
(Event-system_distrusted)
AND
(DATABASE.host =system_distrusted.host)
EXECUTE
<ENCRYPT t1>
<NOTIFY admin>

Similarly to the previous obligation, this obligation is triggered by contextual changes. In this case, one of the systems hosting the database is classified as "distrusted" (because of a virus infection, locally detected spyware, installation of dubious software, etc.) by enterprise monitoring systems. If this event is sent to the obligation management system, this system can trigger the above obligation that will encrypt the data and notify the administrator. Again, this action can be seen as a best-effort, temporary solution to prevent that personal data is compromised.

8) OBLIGATION Oid8:
TARGETS:
t1:< FILE=../audit_log, ATTRIBUTES=(TimeStamp, UserIPaddress, UserName)>
WHEN
(time_counter > time_interval)
EXECUTE
<ENCRYPT t1.UserIpAddress>
<DELETE t1.UserName WHERE t1.TimeStamp<= current_time - 6 months>
<RESET time_counter>

This privacy obligation is defined by a privacy administrator to "purge" the content of an audit log file (for example created by a web server) of specific personal data, as dictated by internal guidelines (for example after six months) and encrypt another portion of the data that can be decrypted later on, in case of need. This is an example of ongoing obligation that is periodically triggered based on a predefined interval of time (for example every week).

All the actions described in the above examples of privacy obligations can (conceptually) be generalised as workflows.

A workflow consists of one or more actions/tasks to be executed, in a specified order. In the remaining part of this thesis the concept of workflow is implied whenever privacy obligation's "actions" are discussed.

Please notice that the language used in the above examples to describe privacy obligations is purely illustrative.

The next chapter will provide more details about the actual representation of privacy obligations along with a description of the architecture of our obligation management system and its implementation.

7 Architecture of Our Privacy Obligation Management System

This chapter provides technical details about the approach and solution we implemented to handle privacy obligations.

7.1 Design Rationale

The design rationale behind our *obligation management syste*m is dictated by the requirements and issues described in Chapter 4 and based on our privacy obligation model and obligation management framework.

As previously anticipated, in our approach privacy obligations are handled in an explicit way independently and not subordinated to access control. This is required in order to deal with privacy obligations that involve deletion of data, notifications or complex workflows, requests for authorizations and executions of workflows that must be triggered independently by access control activities.

Based on this, our design choices reflect the following core aspects:

(1) Privacy obligations are self-standing policies, represented with an appropriate language, separated from access control policies;

(2) The obligation management system must explicitly parse, manage, schedule, enforce and monitor privacy obligations via dedicated modules. In particular, the monitoring of enforced obligations is important to ensure that the overall system is compliant to enforced privacy obligations and that violations are spotted and reported to administrators.

The fact that the obligation management system must handle privacy obligations over long-periods of time and must be always available has also influenced our design choices: survivability and reliability are core requirements. The current design of the obligation management system takes these requirements into account: it is possible to create multiple distributed instances of the obligation management system and monitor for their availability.

7.2 Implementation of Privacy Obligations

Privacy obligations are represented by using an XML format [W3C03d], even if alternative formats are currently under exploration (including RDF [W3C04]).

For the time being, the XML-based format has been chosen as it is suitable for future extensions of the content of privacy obligations, in a modular way.

At the moment the following categories of privacy obligations have been implemented:

- **Transactional obligations**;
- **Short and long term obligations**;
- **Ongoing obligations**.

The events that are currently supported are:

- **Time-based events**;

- **Counter-based events**;
- **Access control-based events for well defined pieces of personal data**;
- **AND/OR combination of the above events**: the AND/OR operators apply to the logical evaluation of events (the fact that they happened means they are TRUE, otherwise they are FALSE) and/or constraints on events. For example, a constraint on a time-based event such as "current_time > Date1" is TRUE if the current time is greater than "Date1" and FALSE otherwise. In this thesis, for brevity, we will refer to "constraints on events" as "events".

The actions that are currently supported are:

- **Deletion of data**;
- **Notification via e-mail**;
- **Triggering of workflow-based actions (external to the obligation management system)**;
- **Sequences of the above actions**.

The .dtd definition of the XML-based format used to represent the above types of obligations follows – Figure 17:

```
!ATTLIST obligation
      oid CDATA #REQUIRED
>
<!ELEMENT obligation (target, metadata, events, actions)>

<!-- target -->
<!ELEMENT target (database)>
<!ELEMENT database (dbname, tname, data)>
<!ELEMENT dbname (#PCDATA)>
<!ELEMENT tname (#PCDATA)>
<!ATTLIST data
      attr (all | part) #REQUIRED
>
<!ELEMENT data (item*)>
<!ELEMENT item (#PCDATA)>

<!-- metadata definition -->
<!ELEMENT metadata (type, description)>
<!ELEMENT type ANY>
<!ELEMENT description ANY>

<!-- events definition -->
<!ATTLIST events
      operator (OR | AND | NOT) #REQUIRED
```

```
>
<!ELEMENT events (event*, events*)>
<!ATTLIST event
      id CDATA #REQUIRED
>
<!ELEMENT event (type, date?, item?, times?, period?)>
<!ATTLIST date
      now (yes | no) #REQUIRED
>
<!ELEMENT date (year, month, day, hour, minute, second)?>
<!ELEMENT period (year?, month?, day?, hour?, minute?, second?)>
<!ELEMENT times ANY>

<!ELEMENT year ANY>
<!ELEMENT month ANY>
<!ELEMENT day ANY>
<!ELEMENT hour ANY>
<!ELEMENT minute ANY>
<!ELEMENT second ANY>

<!-- actions definition -->
<!ELEMENT actions (action*)>
<!ATTLIST action
      id CDATA #REQUIRED
>
<!ELEMENT action (type, data?, method?, to?)>
<!ELEMENT method ANY>
<!ELEMENT to ANY>
```

Figure 17: Privacy Obligation XML Format: DTD definition

Figure 18 shows a simple XML-based privacy obligation, based on the examples described in the previous chapter:

```
<?xml version="1.0"?>
<obligation oid="43459345908605678">
        <target>
                <database>
                        <dbname>oms_demo-customerdb</dbname>
                        <tname>customers</tname>
                        <data attr="part">
                                <item>@key:UserId:uid123|att:creditcard</item>
                                <item>@key:UserId:uid123|att:email</item>
```

```
                    <item>@key:UserId:uid123|att:name</item>
            </database>
        </target>
        <metadata>
            <type>LONGTERM</type>
            <description>
                Delete creditcard AND Notify User
                WHEN current time = 2006:04:19 13:28:00
            </description>
        </metadata>
        <events>
                <event id="e1">
                    <type>TIMEOUT</type>
                    <date now="no">
                        <year>2006</year>
                        <month>04</month>
                        <day>19</day>
                        <hour>13</hour>
                        <minute>28</minute>
                        <second>00</second>
                    </date>
                </event>
        </events>
        <actions>
            <action id="a1">
                <type>DELETE</type>
                <data attr="part">
                    <item>creditcard</item>
                    <item>name</item>
                </data>
            </action>
            <action id="a2">
                <type>NOTIFY</type>
                <method>EMAIL</method>
                <to>email</to>
            </action>
        </actions>
```

```
</obligation>
```

Figure 18: 1st XML-based Example of Privacy Obligation

The content of this privacy obligation is self-explicative. It is about a privacy obligation that targets three fields in a database (i.e. creditcard, name, e-mail) within a database record (associated to a customer), identified by a record key (UserId field, *uid123*). It is a "long-term" obligation, requiring the deletion of the creditcard and name fields at a predefined date and sending a notification of the user via e-mail.

A slightly more complex example of XML-based privacy obligation is shown in Figure 19:

```
<?xml version="1.0"?>
<obligation oid="57856745880978">
        <target>
                <database>
                        <dbname>oms_demo-customerdb</dbname>
                        <tname>customers</tname>
                        <data attr="part">
                                <item>@key:UserId:uid123|att:creditcard</item>
                                <item>@key:UserId:uid123|att:email</item>
                                <item>@key:UserId:uid123|att:name</item>
                                <item>@key:UserId:uid123|att:address</item>
                        </data>
                </database>
        </target>
        <metadata>
                <type>LONGTERM</type>
                <description>
                        Delete creditcard AND Notify user
                        WHEN
                          creditcard has been accessed 2 times
                         OR
                           Either current time is 2006:04:19 13:28:00
                            OR
                            Address has been deleted
                </description>
        </metadata>
        <events operator="AND">
                <event id="e1">
```

```
                    <type>ACCESS</type>
                    <item>@key:UserId:uid123|att:creditcard </item>
                    <times>2</times>
                </event>
                <events operator="OR">
                    <event id="e2">
                        <type>TIMEOUT</type>
                        <date now="no">
                            <year>2006</year>
                            <month>04</month>
                            <day>19</day>
                            <hour>13</hour>
                            <minute>28</minute>
                            <second>00</second>
                        </date>
                    </event>
                    <event id="e3">
                        <type>DELETE</type>
                        <item>@key:UserId:uid123|att:address</item>
                    </event>
                </events>
        </events>
        <actions>
                <action id="a1">
                    <type>DELETE</type>
                    <data attr="part">
                        <item>@key: UserId:uid123|att:creditcard</item>
                    </data>
                </action>
                <action id="a2">
                    <type>NOTIFY</type>
                    <method>EMAIL</method>
                    <to>@key: UserId:uid123|att:email</to>
                </action>
        </actions>
</obligation>
```

Figure 19: 2nd XML-based Example of Privacy Obligation

This privacy obligation requires the deletion of the *creditcard* attribute and the notification of the data subject when one of the composite events happens. This obligation can be triggered when the credit card has been accessed twice or either the data subject's address has been deleted (hence it does not make anymore sense keeping information about the credit card, assuming that acquired goods must be physically delivered) or a specific point of time has been reached.

An example of ongoing XML-based privacy obligations is shown in Figure 20:

```
<?xml version="1.0"?>
<obligation oid="476567765676452456">
        <target>
                <database>
                        <dbname>customerdb</dbname>
                        <tname>customers</tname>
                        <data attr="part">
                                <item>@key:UserId:uid123|att:creditcard</item>
                                <item>@key: UserId:uid123|att:email</item>
                                <item>@key: UserId:uid123|att:*</item>
                        </data>
                </database>
        </target>
        <metadata>
                <type>ONGOING</type>
                <description>
                        Periodically Notify User
                          Every 30 days
                        OR
                          Every time creditcard has been accessed twice
                </description>
        </metadata>
        <events operator="OR">
                        <event id="e1">
                                <type>OGPERIOD</type>
                                <period>
                                        <days>30</days>
                                </period>
                        </event>
                        <event id="e2">
```

```
                    <type>OGACCESS</type>
                    <item>creditcard</item>
                    <times>2</times>
                </event>
        </events>
        <actions>
            <action id="a1">
                <type>NOTIFY</type>
                <method>EMAIL</method>
                <to>email</to>
            </action>
        </actions>
</obligation>
```

Figure 20: 3rd XML-based Example of Privacy Obligation

This privacy obligation dictates that the user must be notified every 30 days (for example that personal data is retained by the enterprise) or every time his/her data is accessed twice.

All the above privacy obligations can be programmatically interpreted and automatically handled by our obligation management system.

The current XML-based syntax of privacy obligations is easy enough to be directly edited and understood by people.

However, graphical tools can be built to automatically generate obligations in the required format, driven by inputs and preferences provided by users. This last approach has been followed when we integrated our obligation management system with an identity management solution (see Chapter 9) in order to simplify end-users interactions and make the underlying mechanisms transparent.

7.3 System Architecture

Figure 21 shows a high-level architecture of an obligation management system supporting the explicit management and enforcement of privacy obligations. This obligation management system consists of the following modules:

- **Obligation Server**: it deals with the authoring, management and storage of obligations. It explicitly manages the association of privacy obligations to confidential data and their tracking and versioning. It pushes active obligations (i.e. obligations to be fulfilled) to the "obligation scheduler". One or more obligation servers can be deployed (and synchronised), depending on needs;

- **Obligation Store and Versioning**: it stores obligations and their mapping to confidential data. Multiple versions of obligations can also be stored in this system, though in the current version of the system this functionality has not yet been implemented;

- **Obligation Scheduler**: it is the module that knows which obligations are active, ongoing obligation deadlines, relevant events and their association to obligations. When events/conditions trigger the fulfilment of one or more obligations, this component activates the correspondent "workflow processes" of the "obligation enforcer" that will deal with the enforcement of the obligation;

- **Obligation Enforcer**: it is a workflow system containing workflow processes describing how to enforce one or more obligations. The enforcement can be automatic and/or could require human intervention, depending on the nature of the obligation. It is extensible via plug-ins, each of them providing a specific enforcement functionality;

- **Events Handler**: it is the module in charge of monitoring and detecting relevant events for privacy obligations and sending them to the obligation scheduler. The detection of events can happen via instrumented application/services. They can also be directly generated by users, administrators, the "obligation monitoring service" and the information tracker;

- **Obligation Monitoring Service**: it is the module, orthogonal to the scheduling and enforcement systems, that monitors enforced obligations by analysing and checking for the effects of their actions i.e. if the personal data targeted by the obligation is in the desired state;

- **Information tracker**: it is a module that focuses on intercepting events generated by data repositories, databases and file systems containing confidential data and providing this information to the event handler. It is aware of the location of confidential data (as described by the obligation policies) and checks for movements and changes happening to this data;

- **Audit Server**: it audits the relevant events and information generated by the overall system modules and involved applications/services;

- **Resource Manager**: it is a module in charge of checking that all the other system components are running and allocating their services to requestors.

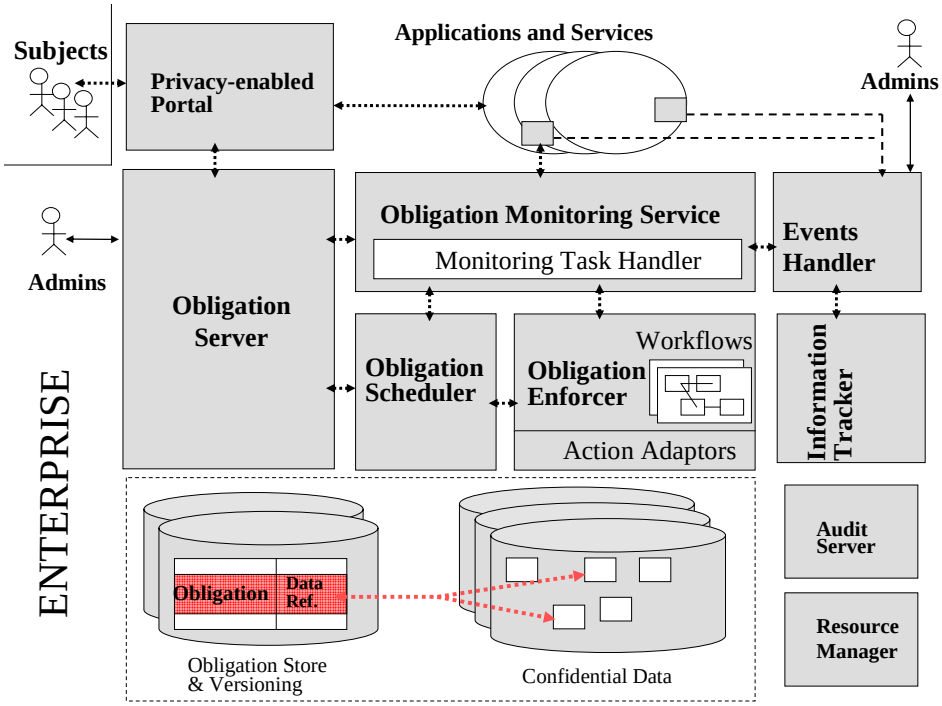


Figure 21: High-level Architecture

The core "run-time" functionalities provided by a system based on this architecture include:

- **Setting a new privacy obligation** (Figure 22): a new obligation is sent to the Obligation Server, either by a data subject or an administrator. The Obligation Server parses and checks for its format correctness. It stores this obligation in the obligation database and communicates it to the Obligation Scheduler to ensure that the obligation will be processed at the due time;
- **Enforcing a privacy obligation** (Figure 23): the Obligation Scheduler listens to managed events sent by the Event Handler and checks if any of them (or any combination of them) triggers one of the managed obligations. Should this happen, the Obligation Scheduler communicates with the Obligation Server to retrieve all the relevant information and sends the obligation to the Obligation Enforcer. The Obligation Enforcer analyses the "action part" of the obligation and executes all the listed actions. Independently by the enforcement result, it sends a copy of the obligation to the Obligation Monitoring Service;
- **Monitoring an enforced privacy obligation** (Figure 24): the Obligation Monitoring Service periodically checks the status of personal data, against related privacy obligations that have been enforced. This is important for compliance reasons, to identify possible violations or technical problems. For example, in case of deleted data (as a consequence of enforcing an obligation) this module will check if data are actually deleted, for a predefined period of time. It might happen that, because of wrong database synchronisation or back-ups, deleted data reappears in the repository: our system will be able to spot this anomaly.

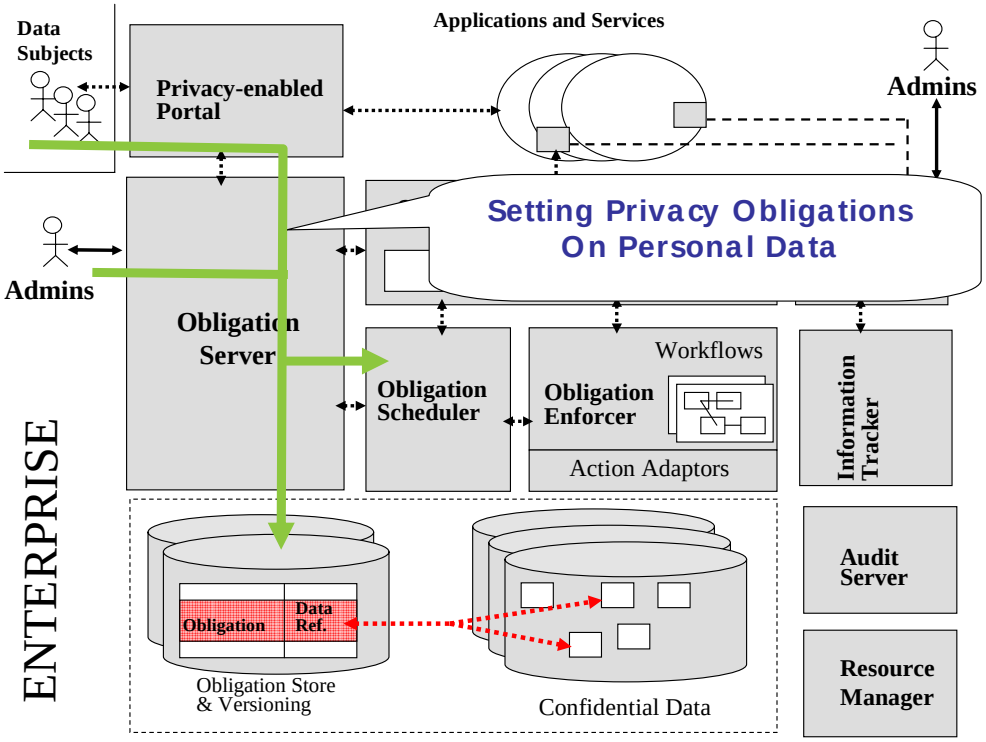


Figure 22: Setting a New Privacy Obligation

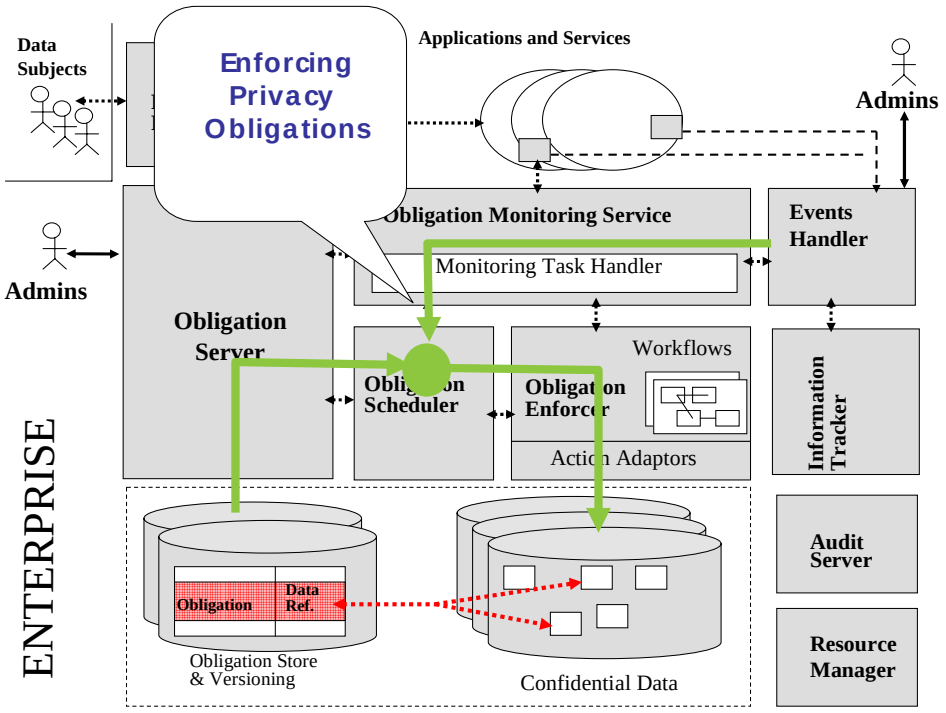


Figure 23: Enforcing a Privacy Obligation

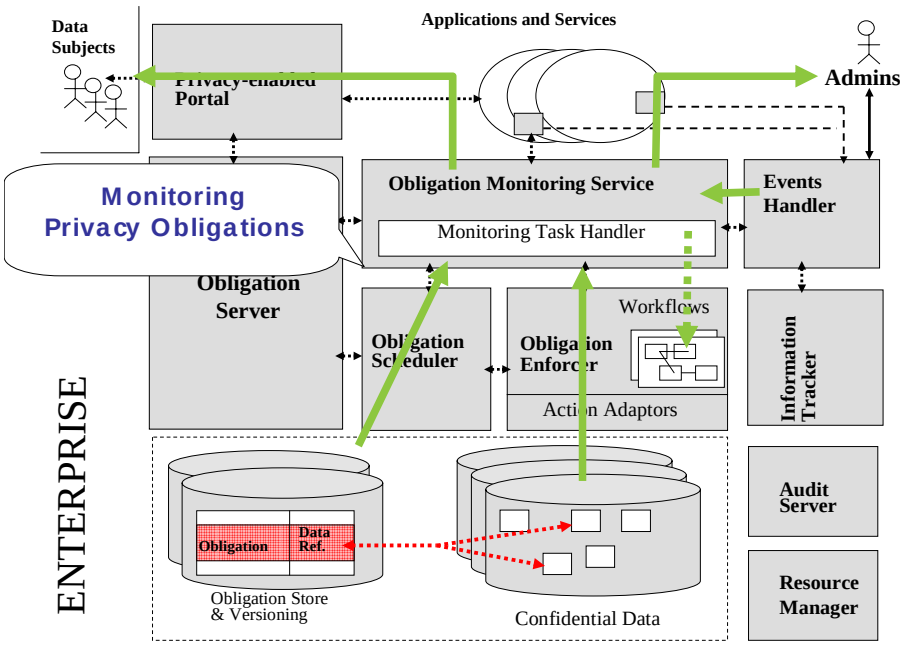


Figure 24: Monitoring a Privacy Obligation

The privacy obligation management system is a critical system: it must survive faults and excessive workloads.

Our system has been built to be distributed and the instantiations of its components can be replicated.

Multiple distributed instances of all the above components can be created and run in parallel: all of them are stateless, as the relevant information on managed privacy obligations is stored in a replicated database. A (replicated) Resource Manager module manages these instances and allocates these resources to requesters (for example the Obligation Server trying to connect to an Obligation Scheduler or the Obligation Scheduler trying to connect to an Obligation Enforcer).

7.4 Prototype: Implementation and Technical Details

This section provides more technical details of the architecture of the obligation management system along with a description of its internal modules, data structures and related interactions, as implemented in a prototype at HP Labs, Bristol, UK in the context of the EU PRIME project [Prim05]. Our prototype has been integrated with external components provided by PRIME partners, in an integrated prototype. Some of our technical choices have been dictated and constrained by this. We will refer to this integrated prototype as the “integrated PRIME prototype”.

7.4.1 Prototype Components

Figure 25 provides a view of the internal modules implemented in the current version of the obligation management system prototype.

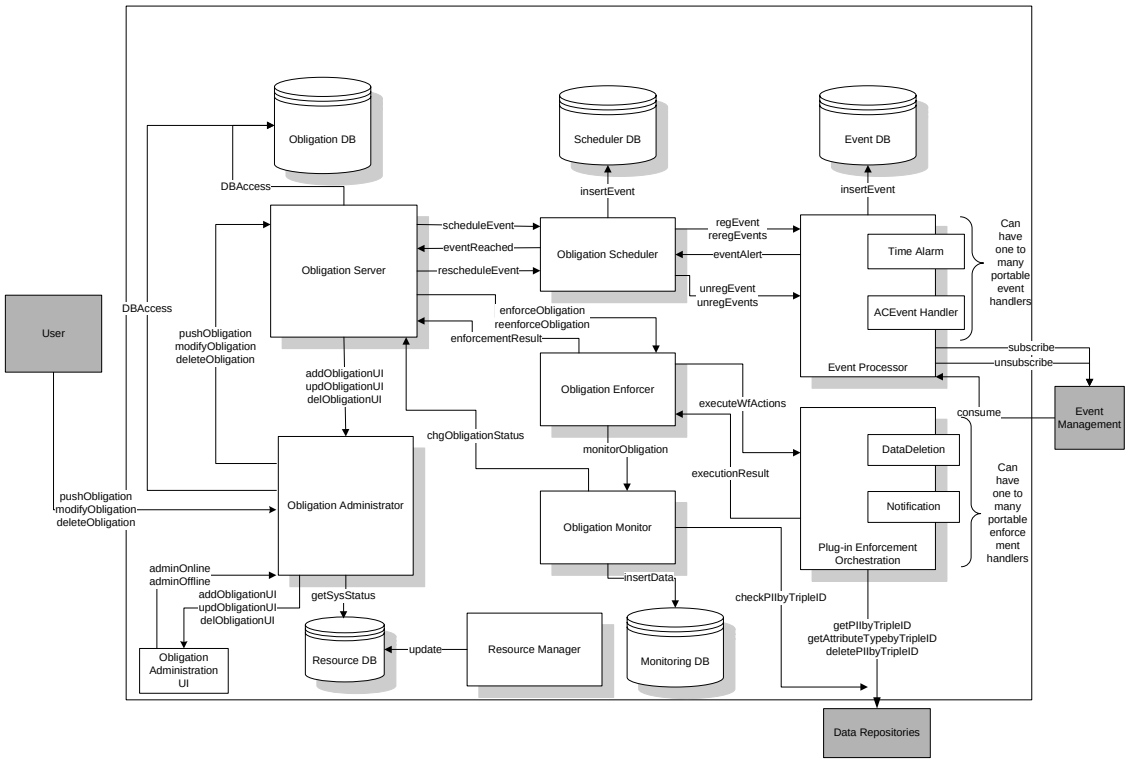


Figure 25: Details of Our Obligation Management System

All these modules have been implemented in Java, as RMI objects. A description of the main modules along with their APIs/interfaces follows.

7.4.1.1 Obligation Administrator

The Obligation Administrator is the module that implements the external-facing functions provided by the obligation management system in order to make them accessible by other external components. It is in charge of interacting with the Obligation Server to coordinate the overall management of privacy obligations within the obligation management system. It also implements internal functions to interact with the Obligation Administration UI and support its management tasks, currently limited to the visualization of active and monitored obligations. The Administration UI mainly interacts with this module. Its key functions are:

pushObligation	This function is invoked by an external component to push a new privacy obligation to the system. It passes this new obligation to the Obligation Server for its processing.
modifyObligation	This function is invoked by an external component to modify an existing privacy obligation in the system. It passes this modified obligation to the Obligation Server for its processing.
deleteObligation	This function is invoked by an external component to delete an existing privacy obligation in the system. It passes this information to the Obligation Server for its processing.

addObligationUI	This function is invoked by the Obligation Server to notify the Obligation Administrator that a new obligation (that has been fully processed by the Obligation Server) must be shown by the Administration UI.
updObligationUI	This function is invoked by the Obligation Server to notify the Obligation Administrator that an updated obligation (that has been fully processed by the Obligation Server) must be shown by the Administration UI.
delObligationUI	This function is invoked by the Obligation Server to notify the Obligation Administrator that a deleted obligation (that has been fully processed by the Obligation Server) must be removed from the Administration UI.
adminOnline	This function is invoked by the Administration UI to notify that it is currently online, i.e. it can receive messages about obligations to refresh the displayed information.
adminOffline	This function is invoked by the Administration UI to notify that it is currently offline, i.e. it cannot receive messages about obligations.

It is important to notice that in the context of the integrated PRIME prototype, privacy obligations (associated to personal data) are also known by other PRIME components, external to the obligation management system. In particular, a copy of these obligations is stored along with personal data in the PRIME "data repository": this because of the data model we chose to implement in PRIME, to guarantee an initial degree of "stickiness" of privacy obligations to data.

Because of this, no "getObligation" method is required to retrieve obligations from the obligation management system as external components can obtain this information directly from the PRIME data repository. This aspect is also reflected by other internal modules of our system.

At the time of writing this thesis, the integrated PRIME prototype also does not support modifications and deletions of privacy obligations. This capability will be provided in a future version of this prototype.

7.4.1.2 Obligation Server

The Obligation Server is the module in charge of processing and handling obligations. It controls all the aspects involving the lifecycle management of privacy obligations: it deals with the local storage of privacy obligations and the coordination of the scheduling of obligations' events and the enforcement of obligations' actions. It also keeps an up-to-date registry of the enforced obligations, based on notifications coming from the Obligation Monitor component. Its key functions are:

pushObligation	This function is invoked by the Obligation Administrator to

	push a new privacy obligation to the Obligation Server. The Obligation Server creates a unique identifier for the new obligation, stores is into the local Obligation Database, notifies the Administrator UI (by invoking the Obligation Administrator's addObligationUI function) about this obligations and interacts with the Obligation Scheduler to set the events relevant to trigger this obligation.
modifyObligation	This function is invoked by the Obligation Administrator to modify a privacy obligation.
deleteObligation	This function is invoked by the Obligation Administrator to delete a privacy obligation.
eventReached	This function is invoked by the Obligation Scheduler to notify the Obligation Server that an obligation has to be enforced, as the events relevant to trigger the obligation happened. The Obligation Server interacts with the Obligation Enforcer to ensure that the relevant obligation's actions are executed.
enforcementResult	This function is invoked by the Obligation Enforcer to notify the Obligation Server about the result of enforcing an obligation. The Obligation Server updates the obligation status within the local database and notifies the Administrator UI.
chgObligationStatus	This function is invoked by the Obligation Monitor to notify that the status of an obligation has changed (i.e. it has been violated or it is OK).

7.4.1.3 Obligation Scheduler

The Obligation Scheduler is the module in charge of scheduling events associated to obligations. These events could be time based (i.e. a specific date and time), access based (i.e. related to access events generated when accessing specific personal data) or based on counters (i.e. the value of an access counter has reached a predefined value). Events could be "ongoing" i.e. they occur periodically (e.g. every month). Events could be composed in logical expressions, involving AND and OR compositions of other events. The current system handles time-based events, counter-based events and ongoing events and their AND/OR compositions. The NOT operator is not yet explicitly supported, as it is not required to handle the current managed set of privacy obligations: it will be introduced in a future version of our system. Its implications on events and complex events need to be fully explored: it is going to be part of our future research activities. This module processes incoming events, forwarded by the Event Processor, and checks if any of them triggers a managed obligation. In case it does, it notifies the Obligation Server, to ensure the enforcement of the relevant obligation. Its key functions are:

scheduleEvent	This function is invoked by the Obligation Server to schedule an event associated to an obligation. This event could be composite i.e. a logical AND/OR logical expression of other events. The Obligation Scheduler parses this event, decomposes it in simple events (if the event is a composite one) and

	stores all this information in the local obligation database. For each simple event it sends related information to the Event Processor, in order to be notified once the event happens. When an event happens, this event is no more considered as active and needs to be rescheduled if its activity needs to be rescheduled.
rescheduleEvent	This function is invoked by the Obligation Server to reschedule an event associated to an obligation. This happens when the obligation is an ongoing obligation, hence events need to be scheduled on an ongoing basis (e.g. once a month). It executes the same activities done by the scheduleEvent function.
eventAlert	This function is invoked by the Event Processor to notify the Obligation Scheduler that a relevant event (previously set by the Obligation Scheduler) has happened. The Obligation Scheduler processes this event and checks if it triggers any managed obligation. In this case, it will interact with the Obligation Server to ensure that the obligation is enforced.

7.4.1.4 Event Processor

The Event Processor module is in charge of processing simple events that are relevant to the obligation management system, to triggering managed events. Based on requests for handling events sent by the Event Scheduler, the Event Processor interacts with any external Event Management component to subscribe (or unsubscribe) for related event notifications (the Event Management component is an abstraction of external components that generate events. Its detailed functionalities are not described as it is beyond the scope of this thesis). This in particular happens for access control-based events or events related to other components. The Event Processor uses a sub-module, called TimeAlarm, to generate time-based events. Its architecture is extensible via plug-in sub-modules, each of them is in charge of receiving and processing specific types of events. In addition to the TimeAlarm plug-in, the current implementation provides an ACEventHandler plug-in, to handle access control related events. The key functions of the Event Processor are:

regEvent	This function is invoked by the Obligation Scheduler to notify the Event Processor about the need to handle a specific type of event. In case of time-based event, the Event Processor will internally register the interest for this event, and at the right time will notify the Obligation Scheduler of its occurrence. In case of access control and other events, the Event Processor will subscribe for this type of events (if it has not yet done it in the past) by interacting to the Event Management component. It will also locally register its interest for this event.
reregEvents	This function is invoked by the Obligation Scheduler to register again for one or more events, in case of ongoing obligations. The relevant events are already known by the Event Processor but some of their parameters might have changed (for example the triggering time, in a time-based event). This function allows the

	Event Processor to update parameters associated to existing event records, to enable the management of ongoing obligations.
consume	This function is invoked by the Event Management component to notify the Event Processor that an event (for which it registered its interest) has occurred. The event passed as a parameter is processed by the Event Processor and sent to the Obligation Scheduler for further processing (such as its evaluation in the context of logical expressions, involving multiple events). It could trigger the enforcement of one or more privacy obligations.
unregEvent	This function is invoked by the Obligation Scheduler to unregister its interest for a particular type of event.
unregEvents	This function is invoked by the Obligation Scheduler to unregister its interest for a set of types of events. This happens in case of ongoing obligations, that have been fully enforced (i.e. do not need to be further processed by the system).

7.4.1.5 Obligation Enforcer

This module is in charge of enforcing privacy obligations i.e. executing actions as defined within privacy obligations once these obligations have been triggered by relevant events. The Obligation Enforcer module is notified by the Obligation Server about the need to enforce obligation actions. These actions might involve the deletion of personal data or part of personal data, sending notifications or handling counters. For ongoing obligations, related actions might need to be periodically enforced (for example resending notifications every month). The Obligation Enforcer interacts with the Plug-in Enforcement Orchestration module to enforce these actions and communicates the outcome to the Obligation Server. It also notifies the Obligation Monitor component about the need of monitoring enforced obligations. Its key functions are:

enforceObligation	This function is invoked by the Obligation Server to notify the Obligation Enforcer module about the need to enforce an obligation, i.e. to execute the actions specified by this obligation. The Obligation Enforcer stores a record in the local database and interacts with the Plug-in Enforcement Orchestration module to execute these actions. It returns the result of the enforcement activity to the Obligation Server.
reenforceObligation	This function is invoked by the Obligation Server to notify the Obligation Enforcer module about the need to re-enforce an ongoing obligation, i.e. to execute again the actions specified by the obligation such as notifications. This might require the system to increase local counters, in case ongoing obligations need to be repeated for a predefined number of times. It returns the result of the enforcement activity to the Obligation Server.
executionResult	This function is invoked by Plug-in Enforcement Orchestra-

	tion module to notify the Obligation Enforcer about the current status of an enforced obligation. Its status could be OK or there could be a FAILURE (obligation enforcement is unsuccessful). In both cases the Obligation Enforcer notifies the Obligation Monitor that it has to monitor the status of this obligation. For example, if the enforced obligation deleted personal data in the database, the Obligation Monitor checks that these data do not reappear in the database.

7.4.1.6 Plug-in Enforcement Orchestrator

This module is in charge of enforcing specific actions as described by a privacy obligation. This might include the execution of complex workflows, involving the coordination of human interactions. Its architecture is extensible via a plug-in based approach. In the current implementation two core actions can be enforced: deletion of data and notification of users via e-mail. In particular for deletion of personal data, it interacts with external data repositories, specifically a RDBMS database. Its key function is:

executeWfActions	This function is invoked by the Obligation Enforcer to notify the Plug-in Enforcement Orchestrator module about the need to enforce one or more actions. In the current version actions might require the deletion of data and notifications to users. The Plug-in Enforcement Orchestrator analyses the types of actions and orchestrates their enforcement by calling plug-in modules, specialized to enforce specific types of actions. At the moment two plug-ins are implemented: DataDeletion and Notification. The DataDeletion plug-in interacts with the data repository to actually delete the relevant data. The Notification plug-in interacts with the data repository to retrieve the actual e-mail address to send a notification to. The overall enforcement result is returned to the Obligation Enforcer.

7.4.1.7 Obligation Monitor

This module is in charge of monitoring enforced obligations for compliance i.e. checking that the effect of enforcing obligation actions is not compromised overtime (e.g. deleted data that reappears in the database because of wrong database back-ups or synchronizations). The obligations that need to be monitored are specified by the Obligation Enforcer. Periodic notifications about the status of monitored obligations are sent to the Obligation Server. At the moment this monitoring capability is passive, in the sense that it highlights violations but it takes no automatic actions to correct it. Administrators need to explicitly ask the system to re-enforce the violated obligations. In a future version of our prototype the automation of this aspect will be further analysed and implemented. Its current key function is:

monitorObligation	This function is invoked by the Obligation Enforcer to notify the Obligation Monitor module about the need to monitor an obligation. It stores a record in the local database about the obligation to be monitored. In case of obligations involving

	deletion of data, it periodically interacts with the data repository to verify if the deleted data has not reappeared.

7.4.1.8 Resource Manager

The Resource Manager is the module in charge of managing, at run-time, the actual RMI instances of all the above modules. The obligation management system ensures the provision of a reliable and survivable service, even in case of occasional, localised failures. To achieve this, runtime redundancy is required for all the above critical components to cope with failures and changing workloads.

Multiple RMI instances of all the above modules can be created at runtime. This is possible as all these modules can run as self-standing RMI objects and all of them are stateless: they store and share all the relevant information within a local database.

In the current configuration, up to three instances of the Resource Manager can run at the same time. Their RMI interface names are well known by all the other modules of the obligation management system: these modules will sequentially try to contact them, until they find a running instance.

At the start-up time, each module registers its RMI interface name to the Resource Manager. In case a module wants to interact with another module, it will first interact with the Resource Manager. The Resource Manager returns the interface name of one of the currently available instances of the requested module.

The Resource Manager also periodically checks for the status of all these instances and updates information in a local database: this information is displayed by the Administrator UI.

The functionalities provided by this component are related to “operational” aspects of the obligation management system and affect all the involved modules: the related functions, described below, are not displayed in the architectural diagram. Its key functions are:

register	This function is invoked by any module of the obligation management system to register its RMI interface name with the Resource Manager. Multiple instances of each module might be registered.
getResource	This function is invoked by any module of the obligation management system to get the RMI interface name of another module of the system. If multiple instances are available and running, the Resource Manager will randomly choose one. If no instance is available, this function will fail.

Our current prototype implements a “synchronised” access to and update of the tables stored in the local databases (“Obligation DB”, “Scheduler DB”, “Event DB”, “Resource DB” and “Monitoring DB”), in order to avoid conflicts and inconsistencies: this is achieved by leveraging standard techniques involving the usage of “critical sections” in the Java code.

7.4.2 Main Interaction Flow

The main interaction flow (involving most of the above modules) is triggered when a new privacy obligation is submitted to the obligation management system. Only the main interaction

steps are described. For simplicity, the description of the steps involving refreshing the UI components is omitted:

1. <pushObligation>: the Obligation Administrator gets a privacy obligation from a user or an administrator. It passes it to the Obligation Server;
2. <pushObligation>: the Obligation Server gets a privacy obligation from the Obligation Administrator;
3. The Obligation Server validates the format of the received obligation;
4. If the obligation is invalid, system returns, process ends;
5. The Obligation Server inserts the valid obligation into the “Obligation DB”;
6. <scheduleEvent>: the Obligation Server extracts the event block from the obligation, sends the event block to the Obligation Scheduler and (asynchronously) waits for the alert confirming that the event has happened;
7. <insertEvent>: the Obligation Scheduler decomposes the complex event into single events, and inserts them into the “Scheduler DB”;
8. <regEvent>: the Obligation Scheduler registers the single events with the Event Processor, and waits for the alert when the event happens;
9. <insertEvent>: the Event Processor inserts the events into the “EventDB”;
10. The Event Processor checks each type of the new events;
11. If the event is time based, it will be sent to the Time Alarm (example of time based event: when the time reaches 01/01/2006 12:00);
12. If the event is access control based, it will be sent to the ACEventHandler that will register its interest in this event with an external Event Management component;
13. <consume>: the Event Processor receives the events from Event Management component. Those events provide the access control information (e.g. the credit card number of user uid05 has been accessed);
14. <eventAlert>: the Event Processor gives alerts to the Obligation Scheduler when the registered event happened;
15. <eventReached>: the Obligation Scheduler updates the status of the event in DB according the received alerts. The Obligation Scheduler sends out the “eventReached” acknowledgement to the Obligation Server when all the conditions in a complex event have been fulfilled;
16. <enforceObligation>: the Obligation Server extracts the action block of the obligation from database, and sends the action block to Obligation Enforcer;
17. <executeWfActions>: the Obligation Enforcer decomposes the complex action into single, ready to enforced actions, and then the actions are sent to the Plug-in Enforcement Orchestration;
18. The Plug-in Enforcement Orchestration forwards the action to suitable plug-ins such as Deletion and Notification plug-ins;
19. <executionResult>: the Plug-in Enforcement Orchestration replies with the execution result;
20. <enforcementResult>: the Obligation Enforcer collects the results from the Plug-in Enforcement Orchestration, then sends the enforcement result to the Obligation Server;
21. <monitorObligation>: the Obligation Server extracts the actions from the obligation, and sends it to Obligation Monitor for monitoring;

22. <insertData>: the Obligation Monitor decomposes the complex actions into single actions, and inserts them into Monitoring DB;
23. <chgObligationStatus>: the Obligation Monitor alerts any violation of the monitoring obligations to the Obligation Server.

It is important to notice that this interaction flow involves steps that can happen in an asynchronous way: for example only when a combination of events happens this triggers the enforcement of related actions.

Further research is required to understand the impact of creating and managing large sets of privacy obligations on large databases of personal data: in this context the management of related events could be critical.

The approach based on replicated instances of critical system components could be exploited to address this issue and balance the workload of the Event Processor.

Further research and work could also be done to optimise the creation and management of privacy obligations, for example by "automatically clustering" obligations that shares the same triggering events, in order to minimise the set of events that must be handled.

7.4.3 Event Management Framework

The obligation management system relies on an external Event Management Framework to receive relevant events and notifications in order to trigger privacy obligations.

As described in the previous sections, the event management model adopted in our system - and pursued in the context of the EU PRIME project - is based on a *producer/consumer* model.

An external event management system is in charge of dealing with registration of producers and consumers and to handle the delivery of generated events.

In this context the obligation management system is just a consumer of events, including:

- Time-based events;
- Access control-based events;
- Intrusion detection events;
- Context-based events (system status, changes of configuration, etc.).

For simplicity, in the current version of the prototype time-based events are directly generated by the "TimeAlarm" sub-module within the "Event Processor" module of our prototype. In a future version, time-based events could also be generated by an external time server and consumed by our system.

As previously stated, it is beyond the scope of this thesis to describe in details what an Event Management Framework is and how it can be implemented. However we recognise that this framework has important implications and requirements on the underlying IT infrastructure.

At the very base, it requires the instrumentation of data repositories, systems and (potentially) applications and services in order to generate the relevant events.

For example, the instrumentation of data repositories, such as RDBMS databases, to generate events based on accesses of stored personal data, might require the definition, deployment and management of triggers and active rules.

In the integrated PRIME prototype [Prim05], that (in addition to our obligation management system) includes various components built by other PRIME partners, the access control system is in charge of intercepting attempts to access personal data stored in databases and (along with making access control decision and enforcing them) generating the relevant events.

Whatever approach is used, an infrastructural overhead is generated. This overhead has to be measured and its impact on the infrastructure and systems has to be quantified.

At the current stage of our work in PRIME, this information is not yet available: this aspect is going to be addressed in a subsequent phase (next 6 months/1 year), once the integrated prototype will be completed and its features fully implemented (including its event management framework).

From an HP Labs perspective, as the obligation management system is orthogonal to the event management framework (as long as it is based on a producer/consumer framework and the semantic of the events is shared with our system), we could leverage event management frameworks already available on the market.

The generation, logging and analysis of events are core functionalities required for IT Compliance Management solutions, in the area of enterprise IT Governance [CaTB05].

Products and solutions available on the market, such as SenSage [Sens05], Synomos [Syno05] and NetForensics [Netf05] provide their own event management frameworks. These frameworks are already deployed in real-world enterprise contexts and a more systematic analyisis of their impacts could be derived from interviews/collaborations with HP customers that use these solutions and/or related case studies.

If compatible with our requirements, the event management frameworks provided by these solutions could be leveraged and integrated with our obligation management system to avoid duplication of efforts.

This aspect and the implications of their integration with our obligation management system will be addressed in a next stage of our project.

7.4.4 Data Repository

For operational reasons the obligation management system stores privacy obligations and related metadata in internal data repositories.

In the previous sections of this chapter we logically referred to these repositories as “Obligation DB”, “Scheduler DB”, “Event DB”, “Resource DB” and “Monitoring DB”.

In the current implementation, for simplicity, all these repositories are implemented as tables within a unique relational database (in our prototype we used a MySQL database system).

The main tables storing this information are:

- **Obligations;**
- **Events;**
- **Expressions;**
- **Actions;**
- **Monitored Items;**

- **Resources.**

A diagram describing the relationships between the above tables is shown in Figure 26:

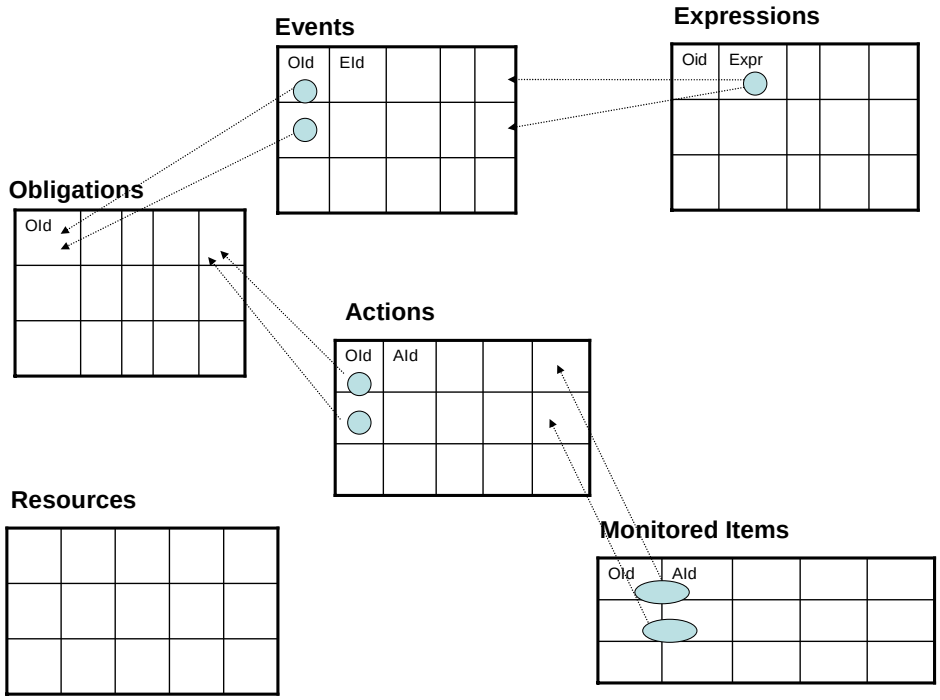


Figure 26: Main Prototype Data Tables

A description of the content of each table follows.

Obligations Table

This table is the main storage of privacy obligations, formatted as XML strings, and related metadata describing their statuses. The main fields of this table are:

ObligationId	it stores the unique identifier of the privacy obligations;
InitTime	it stores the time when this obligation has initially been sent to the obligation management system;
ModifyTime	it stores the last time when this obligation has been modified;
OblType	it stores the type of obligation (long-term, short-term, ongoing, etc.);
Description	it stores a "human readable" description of a privacy obligation, as provided by the administrator that authored the obligation;
Obligation	it stores the entire XML string representing the obligation;
Target	for performance reasons, it stores the "Target" portion of the XML obligation string;
Events	for performance reasons, it stores the "Events" portion of the XML obligation string;

Actions	for performance reasons, it stores the "Actions" portion of the XML obligation string;
Status	it stores the current, up-to-date, status of a privacy obligation (scheduled, enforcing, ok, violated).

Events Table

This table contains a list of "simple events" associated to privacy obligations managed by the system. Its content is the result of parsing the "Events" section of each obligation. As a result, multiple event records could be associated to the same obligation. The main fields of this table are:

ObligationId	it is the unique identifier of the obligation an event belongs to;
EventId	it is the unique identifier of an event, in the context of an obligation;
EventType	it classifies the type of managed event (timeout, access, delete);
ScheduledNumber	it contains the number of times this events is expected to happen to trigger the obligation. It is a counter;
Status	it contains the current status of the event (stopped, closed, etc.);

Expressions Table

This table refers to events contained in the "Events" table and explicitly describes logical combinations (AND, OR combinations) of these simple events. These logical combinations are derived from the original "Events" sections of privacy obligations:

ObligationId	it is the unique identifier of the obligation an event belongs to;
Expression	it is a string containing a logical combination of simple events. Multiple simple events, defined in the "Events" table, are combined in AND/OR logical expressions, by using their EventId;
ScheduledNumber	it contains the number of times this complex event is expected to happen to trigger the obligation. It is a counter;
Status	it contains the current status of the complex event (stopped, closed, etc.);

Actions Table

This table contains a list of "simple actions" associated to privacy obligations managed by the system. Its content is the result of parsing the "Actions" section of each obligation. As a re-

sult, multiple actions could be associated to the same obligation. The main fields of this table are:

ObligationId	it is the unique identifier of the obligation an action belongs to;
ActionId	it is the unique identifier of an action, in the context of an obligation;
Action	it contains the XML portion describing this action (delete, notify, trigger workflow, etc.);
EnfNumber	it contains the number of times this action has been enforced;
Status	it contains the current status of the action (success, failure, etc.);

The ActionId key is relative to the context of an obligation and unique only within this obligation (i.e. the same key could be used in different obligations). The combination of the ObligationId and ActionId keys ensures the unique identification of an action, within the obligation management system.

MonitoredItems Table

This table contains the status of enforced obligations. Specifically the system stores the status of each action of a given enforced obligation. The main fields of this table are:

ObligationId	it stores the unique identifier of the privacy obligations;
ActionId	it is the unique identifier of an action, in the context of an obligation;
InitTime	it stores the time when this obligation has initially been sent to the obligation management system;
ModifyTime	it stores the last time when this obligation has been modified;
Action	it contains the XML portion describing this action (delete, notify, trigger workflow, etc.);
Type	it describes the type of enforced action (delete, notify, etc.);
Status	it stores the up-to-date status of an enforced action (ok, violated).

It is important to notice that also in this table both the ObligationId and the ActionId keys are used to identify an action, for the reasons explained in the “Action Table” subsection.

Resources Table

This table contains the information about all the instances of RMI modules of the obligation management system and their statuses. The main fields of this table are:

Module	it contains the type of system module (ResourceManager itself, ObligationServer, ObligationScheduler, ObligationEnforcer, ObligationMonitor, EventProcessor, EnforcementOrchestrator). Multiple records of the same type could be present, as the system can handle multiple instances of the same components, for fault tolerance and load balancing reasons;
Server	it contains the logical (DNS) name of the server hosting this module;
RMIName	it stores the RMI logical name of the module, used by other module to remotely connect to the object;
Status	it stores the current status of the module (alive, dead).

7.4.5 Administration UI

The current prototype provides (basic) administrative management functionalities via a graphical Administrative UI. This UI provides the following graphical views:

- **Admin View**;
- **Monitoring View**;
- **System View**;

In the **Admin View** of this UI administrators can check and browse for the current set of managed privacy obligations (either to be enforced or enforced) – see Figure 26. In this context, it is possible to restrict, in a fine grained way (based on time intervals), the set of privacy obligations that an administrator wants to investigate.

For each managed obligation, the UI provides the following information:

- **Obligation Id**: it is the unique privacy obligation identifier, used within the entire system;
- **Initialization Time**: it is the time when the privacy obligation has been initially submitted to the system;
- **Modification Time**: it is the last time recorded where the privacy obligation has been subject to any management activity;
- **Type**: it is the type of privacy obligations. The current supported types are: “SHORT-TERM”, “LONG-TERM”, “TRANSACTIONAL”, “ONGOING”;
- **Status**: it describes the current, up-to-date, status of the obligation. The current supported statuses are: “SCHEDULED”, “ENFORCING”, “OK”, “VIOLATED”;
- **Description**: it is a human readable description of the privacy obligations. This information is derived from the metadata associated to the privacy obligations, within its XML format.

It is important to notice that this UI can provide a list of all the managed privacy obligations whatever their statuses are. However, because this list can be very large, it could be unmanageable. The current UI already provides the administrators with mechanisms to focus on a subset of this list, based on any combination of the following criteria:

- Initialization time of an obligation;
- Modification time of an obligation;
- Status on an obligation.

These filtering mechanisms are made available to the administrators via a few selection fields, available at the bottom of the UI - see Figure 27.

By double-clicking on any obligation row, the administrator can get a detailed view of the internal components of this obligation, via a pop-up window. This window contains a tree-based representation of the obligation that can be easily navigated – see Figure 27.

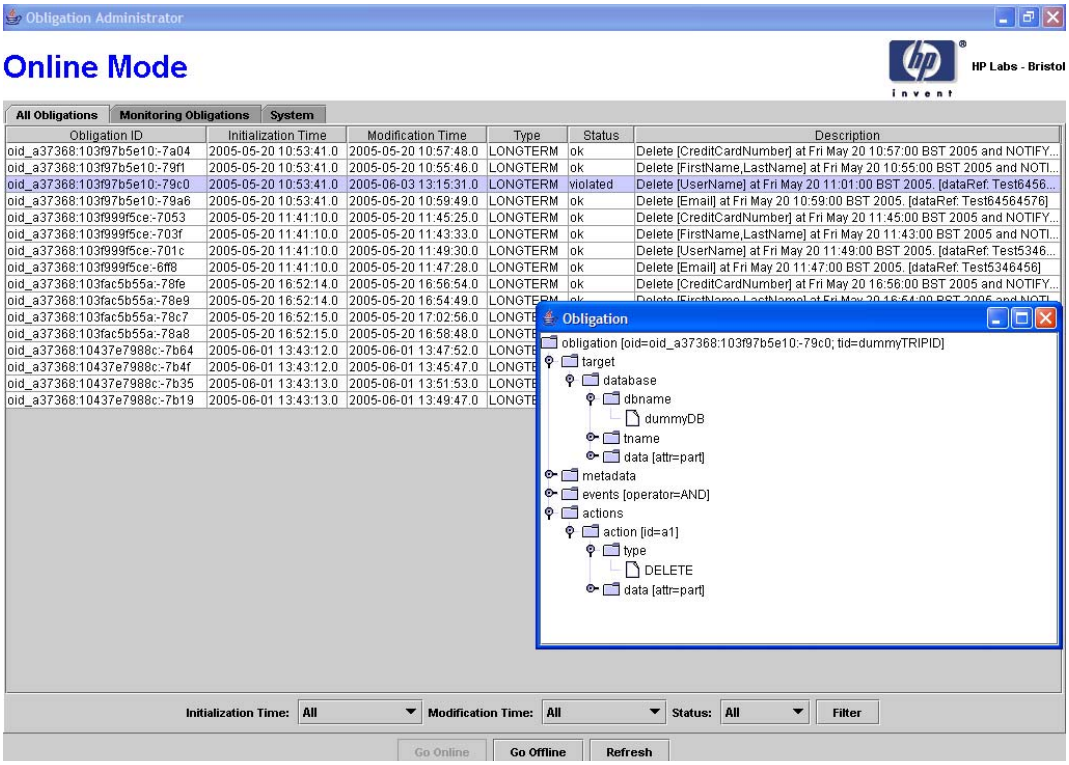


Figure 27: Obligation Management: Administrative UI

The **Monitoring View** is based on a similar UI, with exactly the same fields. However this UI provides a graphical view of the status of privacy obligations that have been enforced and that are currently monitored – see Figure 28.

Each obligation is displayed with an associated colour:

- **GREEN**: the status of the obligation is OK. This means that the data targeted by the obligation is in the expected status, as dictated by the enforced obligations;
- **RED**: the obligation is VIOLATED. This means that that the data targeted by the obligation is not in the expected status, dictated by the enforced obligations.

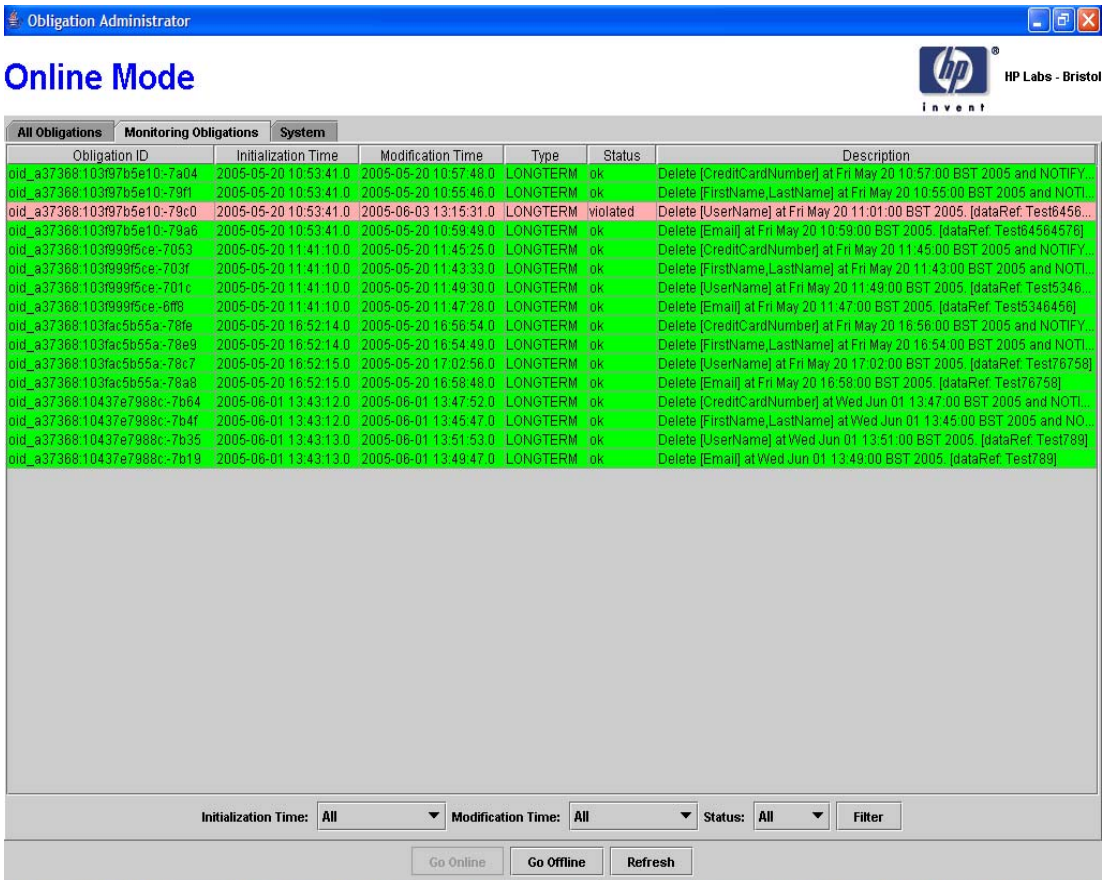


Figure 28: Obligation Monitoring: Administrative UI

The **System View** provides a system perspective illustrating the current status and availability of the various system modules – see Figure 29.

For each system module the UI shows the following information:

- **Component**: it is the logical name of a system module (e.g. Resource Manager, Obligation Server, Obligation Enforcer, Obligation Monitor, Event Processor, Enforcement Orchestration). More than one instance of the same name could appear, as each of these module might be instantiated multiple times, for fault tolerance and load balancing reasons;
- **Server**: it is the name of the server (platform) hosting the instance of the RMI module;
- **RMI Name**: it is the name of the RMI interface associated to the module;
- **Status**: it provided an up-to-date status of the module (e.g. “DEAD” or “ALIVE”).

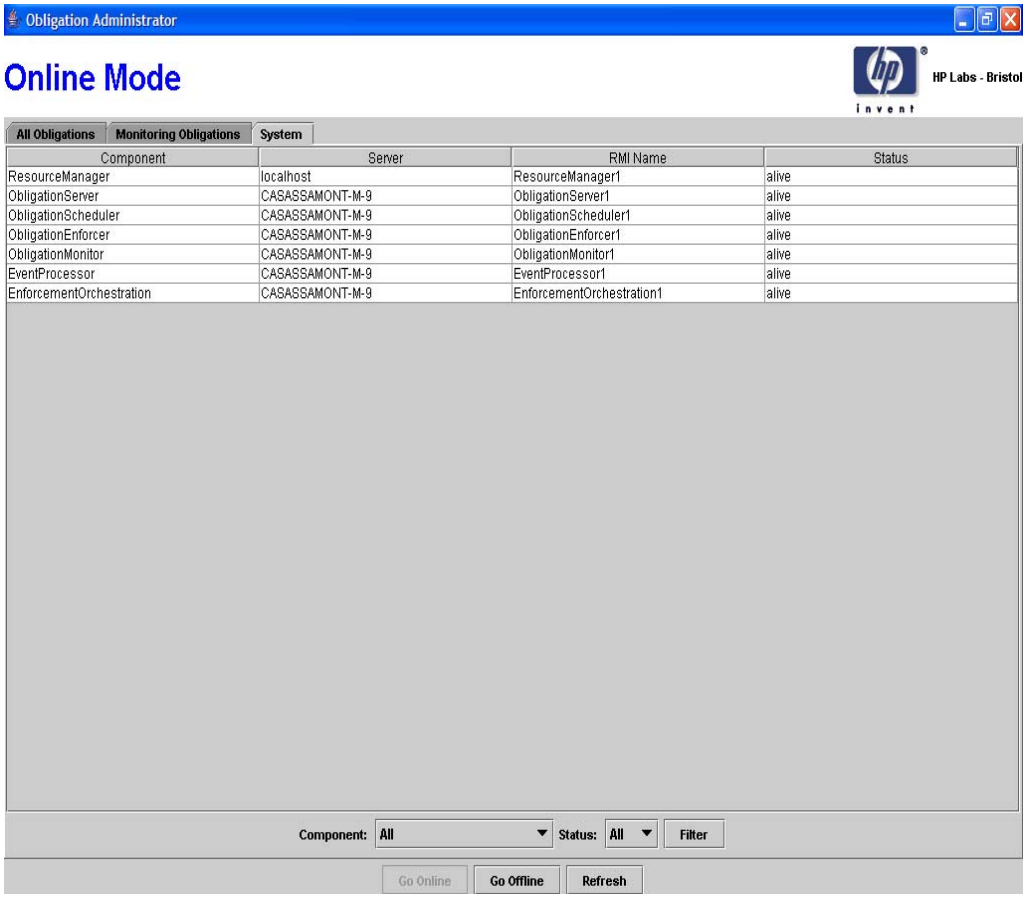


Figure 29: System Component Monitoring: Administrative UI

8 Scenarios and Use Cases

In this chapter we analyse different scenarios where the obligation management system can be deployed and related use cases.

8.1 Scenario: User Provisioning

In this scenario, the obligation management system is deployed in an identity management context and used to handle privacy obligations. Privacy obligations are defined at the time of user self-registration and provisioning – see Figure 30.

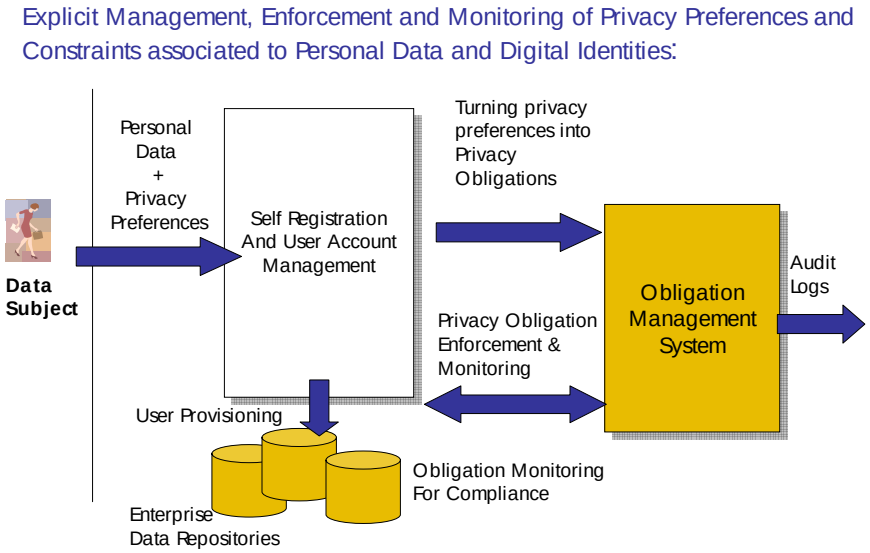


Figure 30: Use Case 1 - User Provisioning

The following interactions apply:

- A new user decides to self register via an enterprise or organisation's web portal;
- The self-registration process requires the user to provide data, including personal data. In this step the user can also specify their privacy preferences (such as constraints on deletion of these data or notifications);
- The self-registration module triggers the provisioning of the user (creation of user accounts, setting of access control rights and storage of personal information) to the various involved enterprise systems;
- During the user provisioning process, privacy obligations are generated from user's privacy preferences and user's personal data. This happens by creating, on-the-fly, privacy obligations where the target is the personal data and the events and actions are dictated by the expressed preferences. These obligations are pushed to the obligation management system;
- The obligation management system takes care of scheduling, enforcing and monitoring these privacy obligations.

In this scenario, typical privacy obligations derived from user's preferences might include the deletion of personal data after a predefined period of time and/or require explicit requests for notifications and authorizations, for example when these data are accessed or disclosed to third parties.

8.2 Scenario: Privacy-aware Management of Personal Data

In this scenario, the obligation management system is a standalone system in charge of dealing with a privacy-aware lifecycle management of personal and confidential data stored by an enterprise – see Figure 31.

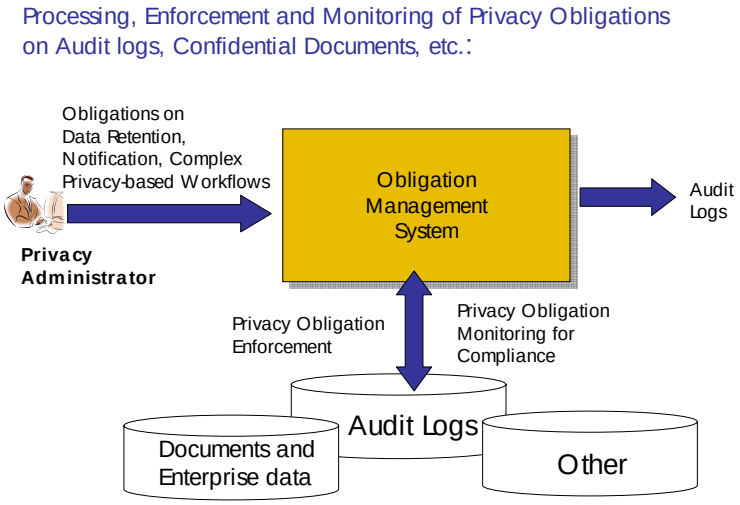


Figure 31: Use Case 2 – Privacy-aware Management of Confidential Information

Personal and confidential data can be stored in a variety of data repositories, log files, audit systems, etc.

Privacy administrators specifically define and push privacy obligations (related to the above types of data) to the obligation management system that will take care of scheduling, enforcing and monitoring these privacy obligations.

In this scenario, the obligation management system can be used to periodically purge audit log files (generated by various enterprise systems, such as web servers), re-format data, minimize (via encryption or deletion) some of the personal data or statistically transform these data.

8.3 Scenario: Management of Complex Workflow

This scenario is complementary to the previous two scenarios.

Privacy obligations can be used to define actions that involve the execution of complex workflows requiring activities on personal data and sequences of human and computer interactions (such as requests for authorization to the responsible people).

The obligation management system can be used as a component to schedule these privacy-aware data lifecycle management workflows, execute them at the right time and monitor for their enforcement – see Figure 32:

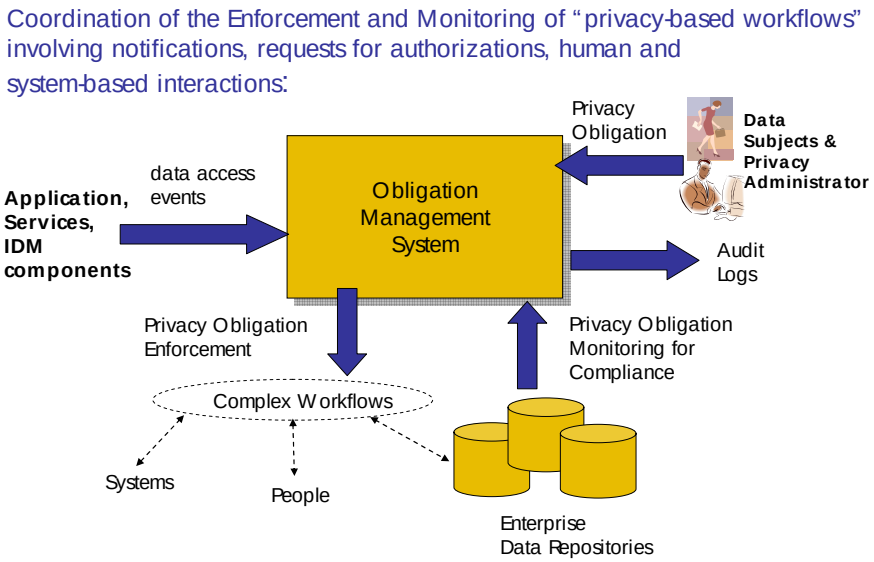


Figure 32: Use Case 3 – Management of Complex Workflows

9 Deployment of Our System in a Real-world Identity Management Solution

To demonstrate the actual feasibility and practicality of our approach we investigated how our current obligation management technology can be integrated and leveraged by state-of-the-art identity management solutions. This is an important requirement as enterprises are currently investing on identity management solutions to handle the lifecycle of digital identities and user accounts.

Figure 33 illustrates the high level functional architecture of state-of-the-art identity management solutions [CaBP03], along with the logical collocation of our privacy obligation management system (OMS).

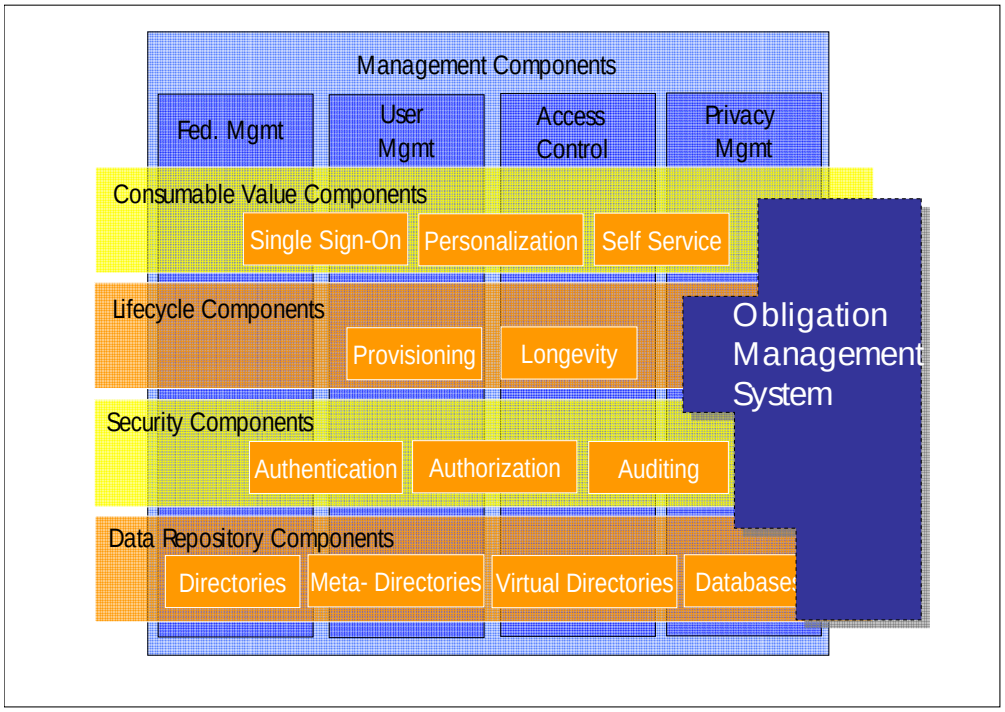


Figure 33: Current Identity Management System and Integration with OMS

Our obligation management system spans across different functional components of the identity management solution, as it has to:

- interact with the self-service and provisioning capabilities, during the provision and management of new identities and personal data;
- deal with authentication, authorization and auditing issues;
- interact with various data repositories storing personal data that are subject to the managed privacy obligations;

The integrated system must allow data subjects to define as early as possible their privacy preferences, during their self-registration phase, at the time their personal data are disclosed. The user provisioning and account management components must be able to process these

preferences and turn them into privacy obligations to be managed by our privacy obligation management framework.

We specifically focused on the self-registration and user provisioning scenario, as described in the previous chapter. This scenario is relevant as it allows data subjects (users) to be directly involved in the process, by specifying their privacy preferences on their personal data and, at the same time, it allows leveraging the provisioning functionalities provided by modern identity management systems.

9.1 Integrated Prototype

We integrated our obligation management system prototype (developed in the context of PRIME [Prim05]) with HP Select Identity.

9.1.1 HP Select Identity

HP Select Identity [Hewl05a] is a state-of-the-art identity management solution to manage digital identities and user accounts within and between large enterprises. It automates the process of provisioning, managing and terminating user accounts and access privileges across platforms, applications, and corporate boundaries. Specifically, the key features of the Select Identity system include:

- **Centralized Management**: provides a single point of control for the management of users and entitlements;
- **Provisioning**: automates the creation, update, and deletion of accounts and entitlements on information systems across the enterprise. This happens by interfacing to these systems via connectors (for example there are connectors to RDBMS databases, LDAP directories, etc.). Connectors contain the logic and the knowledge of how to handle data and execute operations on these systems. New connectors can be specifically built for new systems or a legacy systems;
- **Administrative Delegation**: enables administrative rights to be distributed among multiple tiers of functional departments, customers, and partners;
- **User Self Service**: enables end users to initiate access to Services, change passwords, set password hints, and update general identity information through a web browser interface;
- **Approval Workflow**: automates approval processes required for granting access privileges to users;
- **Password & Profile Management**: manages and distributes password and user profile information across and between enterprise information systems;
- **Audit and Reporting**: provides standardized reporting on actions and user account activity.

Figure 34 provides a high level overview of HP Select Identity's architecture and how the above functionalities are mapped in architectural components.

It is important to notice that HP Select Identity's functionalities are accessible both from an administration UI (by administrators) and programmatically, via a web service API.

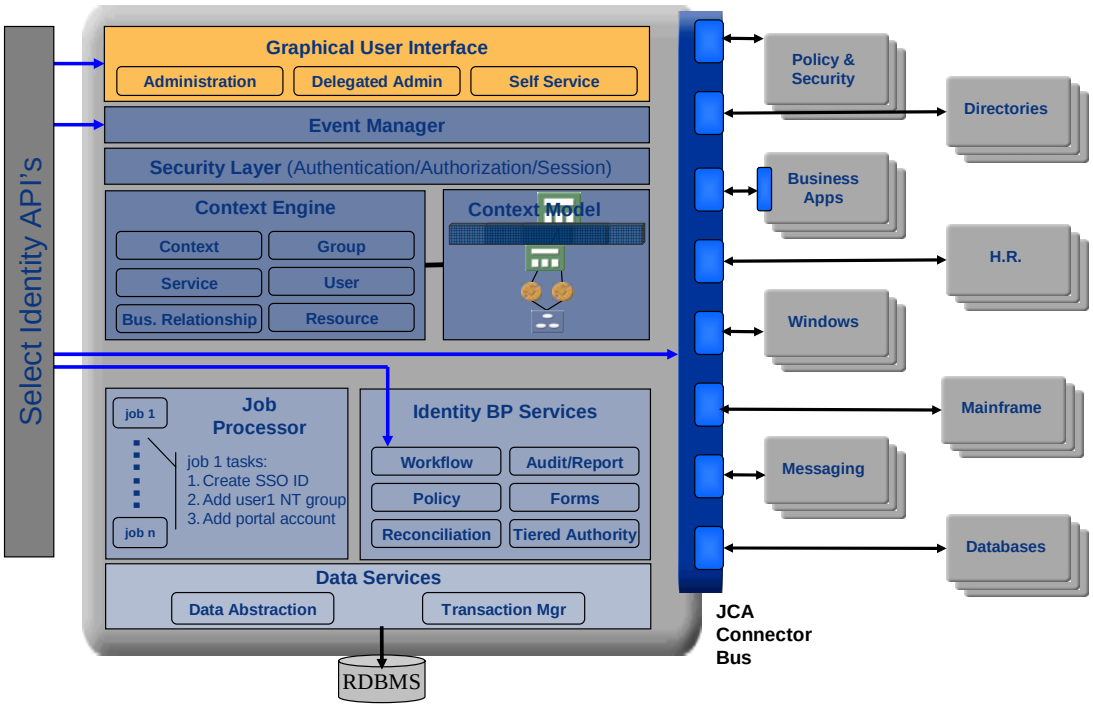


Figure 34: HP Select Identity Architecture

A set of connectors (Java JCA adaptors) are currently available to enable HP Select Identity to interact with a variety of third party components, including common data repositories, windows OS a few common business applications. For more details see [Hewl05a].

These connectors can be used to:

- Provision data (and accounts) to an external data repository or system;
- Report back to HP Select Identity about any change that happened on data provisioned to external repositories and systems. This happens via an "Agent-based" mechanism that is activated by data changes (for example via triggers, in databases): it will report these changes to HP Select Identity via its Web Service API.

New JCA connectors can be built and deployed in HP Select Identity to interact with specific systems and applications.

HP Select Identity maintains a local, up-to-date copy of all the identity information it has provisioned to external systems, along with related metadata (i.e. which data have been provisioned to which external systems, required transformations of data, etc.): this is required for its internal processing activities. This repository will be referred as the "virtual identity repository".

9.1.2 Integration Details

In our integrated prototype, shown in Figure 35, we leverage HP Select Identity self-registration and user provisioning capabilities to capture users' privacy constraints and preferences on how to handle personal data.

Currently the prototype can handle deletion of information associated to users - in a fine grained way, at the level of identity attributes or the entire user profile - and notification preferences.

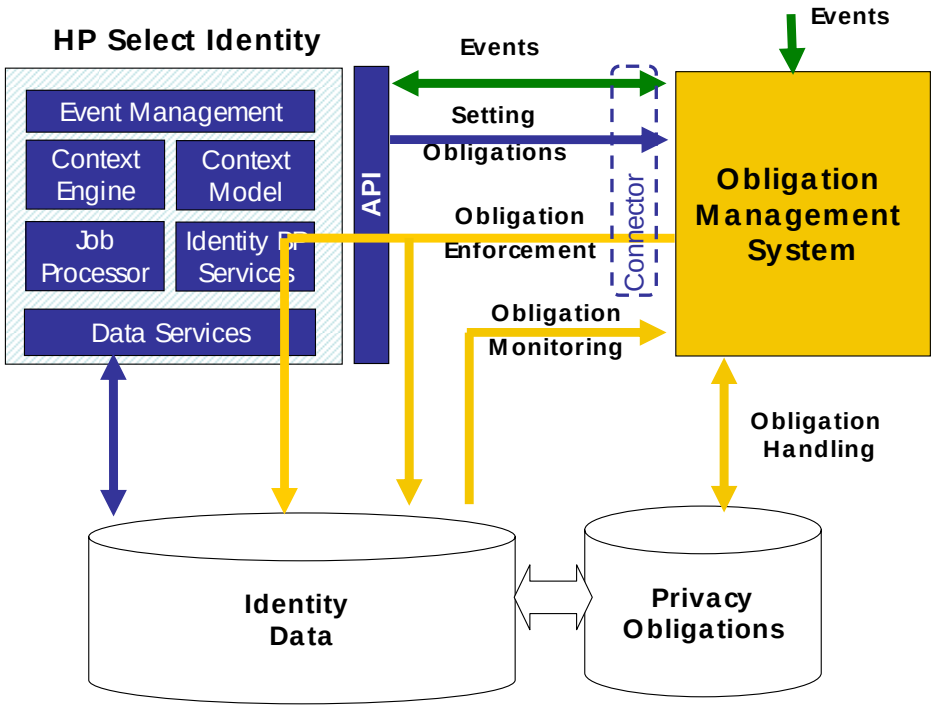


Figure 35: Integration of our OMS system with HP Select Identity

In this prototype, a new connector has been built to connect HP Select Identity to the obligation management system. Our system is perceived by HP Select Identity as being yet another external data repository.

Privacy preferences are detected and processed by a module (Obligation Translator) provided by this connector and transformed into privacy obligations. Privacy obligations are scheduled, enforced and monitored by our obligation management system. We leverage the workflow and user/identity management capabilities of HP Select Identity to enforce aspects of privacy obligations. These functionalities are accessed by our system via the web service APIs provided by HP Select Identity.

Specifically, HP Select Identity is used to enforce obligations constraints, such as deletion of identities, data transformation, etc. At the moment the deletion of personal data is achieved by triggering HP Select Identity workflows, whilst the obligation management system handles the notifications to users.

Our obligation management system also retains control of the supervision of obligations and their monitoring.

A more detailed description of the integrated prototype and the flow of the involved interactions are shown in Figure 36:

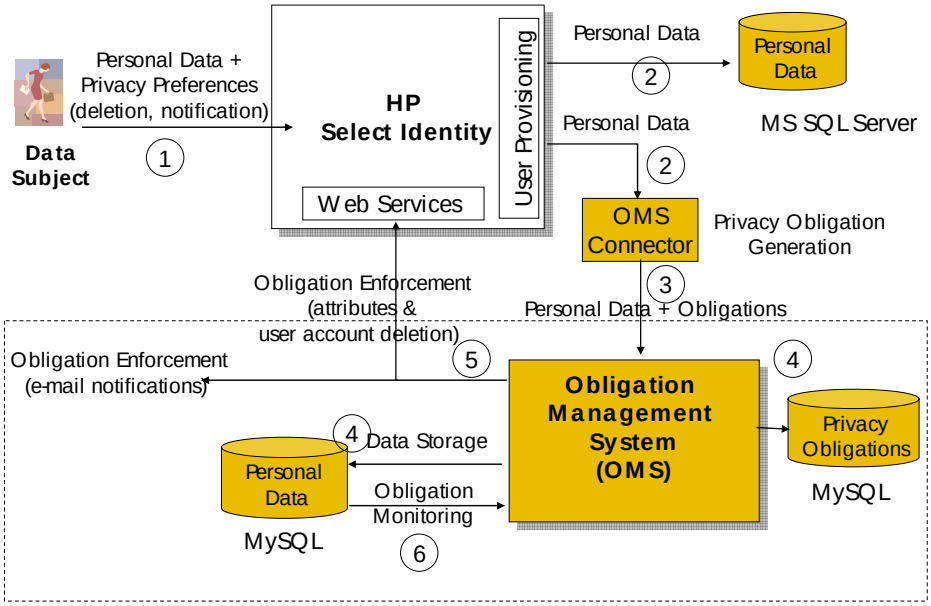


Figure 36: Integration of our OMS system with HP Select Identity - Details

The following basic interactions/steps happen:

1. A new user accesses HP Select Identity self-registration portal, in order to access enterprise web services. He/she provides her personal information along with privacy preferences and submits the request;
2. HP Select Identity processes this request and makes a decision of which data repositories and systems are affected by the provisioning of this new user. In the current prototype there are two systems affected:
    a. **A RDBMS repository**: personal data and preferences are simply stored there via a MS SQL connector;
    b. **Our Obligation Management System (OMS)**: the OMS system is simply seen by HP Select Identity as another data repository. The OMS connector (we built) mediates interactions with the OMS system;

    This provisioning process, managed in our prototype, is very simple: it is a proof of concept, aiming at demonstrating the feasibility of our work. In a real-world scenario, HP Select Identity might have to provision different enterprise data repositories (e.g. LDAP directories, RDMS systems, meta-directories, legacy repositories) and external applications/solutions (e.g. SAP, etc.). It is beyond the scope of this thesis to describe in details how HP Select Identity and its connectors can be used to achieve this. More information is available at [Hewl05a].
3. The OMS connector, once invoked by HP Select Identity, processes the incoming data, in particular the privacy preferences set by the user. These privacy preferences are automatically turned into privacy obligations (in the XML format described in

Chapter 7) by an internal module, called Obligation Translator and pushed to the Obligation Management system;

4. The OMS system processes the incoming privacy obligations, stores a copy of them in the local obligation database and schedules them for their enforcement. In this prototype a local copy of the relevant personal data (targeted by these obligations) is stored in the OMS local database to enable, later on, the monitoring of privacy obligations.

   The reason for doing this is based on the assumption that any modification of personal data (i.e. modifications, deletions) made on *external* data repositories (for example the external RDBMS system) will trigger a reaction of HP Select Identity to ensure the *realignment* with all the other provisioned copies of these data.

   Please notice that this also covers the case of an accidental restore of deleted data, due to the usage of the wrong backup file or because of wrong database synchronisations. As a consequence, HP Select Identity will realign the data stored on all the provisioned system (despite the fact these changes are due to a mistake).

   This realignment can be achieved by instrumenting the external repositories and systems - by means of triggers in databases or ad-hoc reporting mechanisms, embedded in specific connectors - to notify HP Select Identity of these changes. As a side-effect, HP Select Identity will propagate changes to the other provisioned systems, including the OMS. This will also affect the OMS local copy of the data (our OMS connector is able to understand this situation).

   Hence the OMS can monitor its local copy of the data to verify the status of enforced obligations. The enforcement of privacy obligations provokes itself changes of stored personal data, such as their deletion (this process is initiated by the OMS, invoking the HP Select Identity web services' API).

   We recognise the limitations of this approach: we adopted it for a matter of quick prototyping, as a proof of concept. A better and more robust approach would have consisted in using the HP Select Identity's web services' APIs to check the status of its managed data, as stored within its "virtual identity repository" (that, by definition, should contain an up-to-date version of all its managed data);

5. At the enforcement time of a privacy obligation, the OMS system triggers the execution of relevant actions. In the current prototype, this might involve:

   a. Deleting a user account or attributes of a user profile: this is done by the OMS system by invoking relevant web service APIs provided by HP Select Identity. As a reaction, HP Select Identity will execute the required commands on all the data repositories storing the affected data. As a side effect, also the OMS local data repository (see above) is affected by these changes;

   b. Notifying a user about the deletion of his/her data: this is directly managed by the OMS system, by sending e-mails to the user;

6. The OMS system carries on monitoring for the enforced obligations: the actual deletion of personal data is checked by controlling the status of the local data repository (that should be affected by changes made by HP Select Identity). In the current prototype, this is based on the assumptions described at point 4.

The core interaction flows work as planned and we can effectively handle simple privacy obligations (based on notification and deletion preferences) on personal data managed by HP Select Identity.

Of course, this is a proof of concept and currently it is work in progress. The current prototype is going to be extended in order to handle more complex privacy obligations and fully leverage the workflow capabilities provided by HP Select Identity.

Further work is required to understand the implications (and limitations) of dealing with modifications of personal data via HP Select Identity in a broad (real-world) variety of data repositories and external systems, in particular in terms of the instrumentation (in terms of notification of changes) required to ensure that the OMS system can have an accurate perception of the status of its enforced privacy obligations.

In case of limitations of the current approach, a hybrid approach could be pursued, where part of the checking of the status of managed data (as expected by associated privacy policies) could also be done by OMS Monitoring "plug-ins" built by us.

9.1.3 Integrated Prototype: Demo Snapshots

This section contains a few snapshots of the obligation management system integrated with HP Select Identity. Figure 37 shows the graphical UI of the two main involved components: HP Select Identity and our obligation management system:

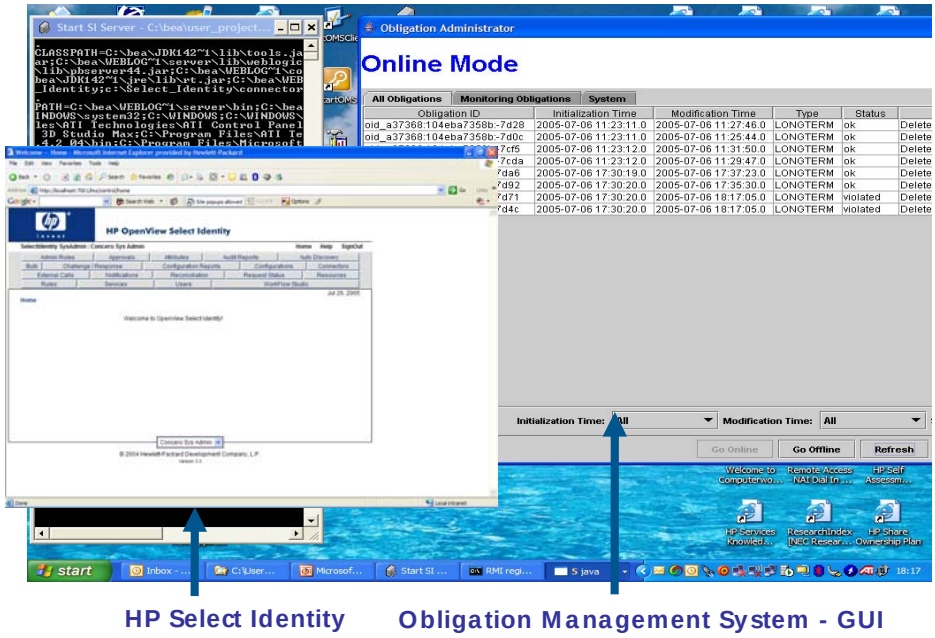


Figure 37: Demo Environment

Figure 38 and Figure 39 show how a new user can self-provision to a web portal by providing personal information (i.e. attributes such as name, surname, creditcard, e-mail, etc.). In this example, the user, in addition to this, can specify related privacy preferences. For each attrib-

ute and/or for the entire user data the user can specify if he/she wants to delete the attribute at a specific point in time and be notified about the success of this event:

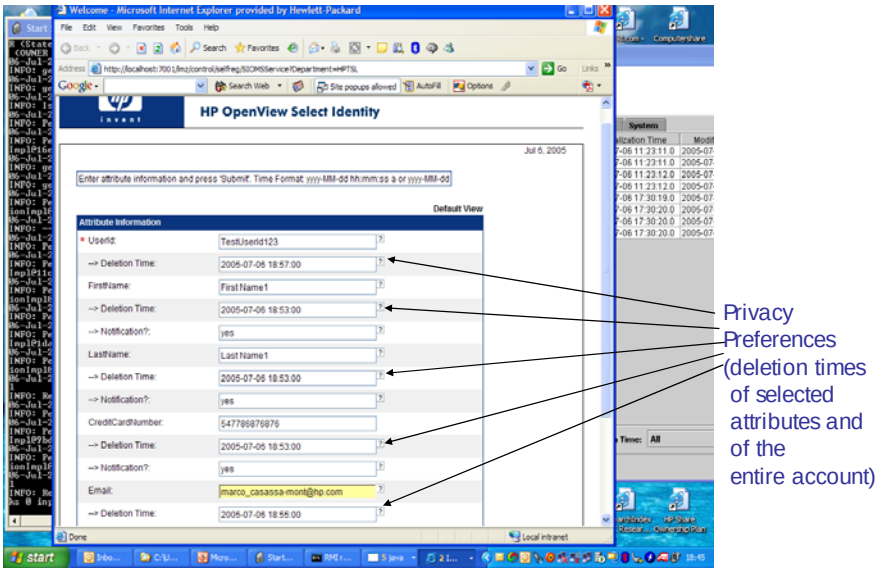


Figure 38: Self-Registration – User's Specification of Deletion Preferences

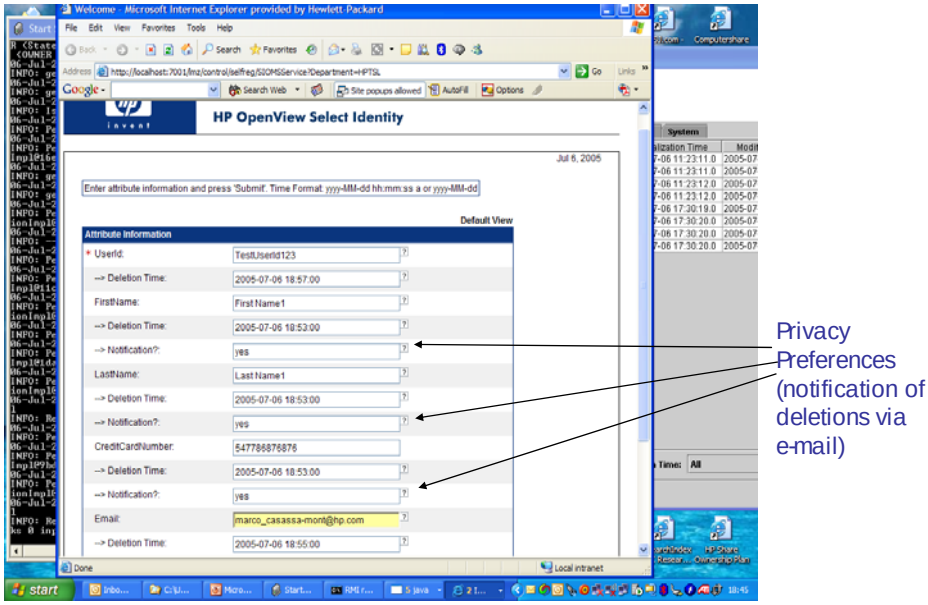


Figure 39: Self Registration – User's Specification of Notification Preferences

Figure 40 shows HP Select Identity confirming that the submission of the new user information has successfully completed and its provisioning is currently planned. HP Select Identity

internally calculates how to provision the user's data to the various data repositories, via their registered connects. This includes the obligation management system, connected to HP Select Identity via the OMS connector.

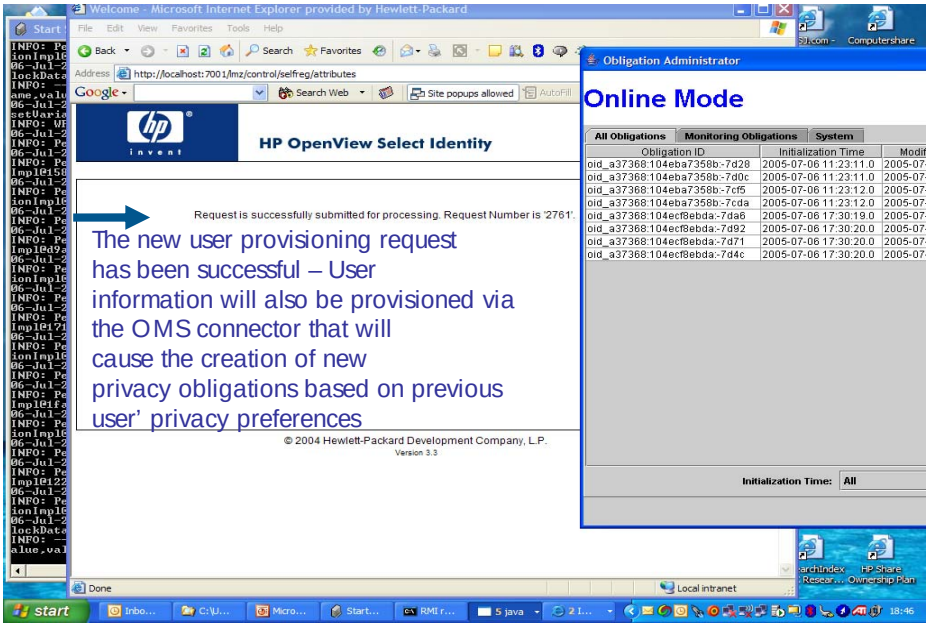


Figure 40: Starting the Provisioning of a New User

The provisioning process will eventually interact with the OMS connector. Based on the data and the user's preferences, new privacy obligations are generated and pushed to the obligation management system. Figure 41 shows the instant where three new privacy obligations are created, as an effect of the user provisioned in the example.

Figure 42 provides more details about the newly generated obligations.

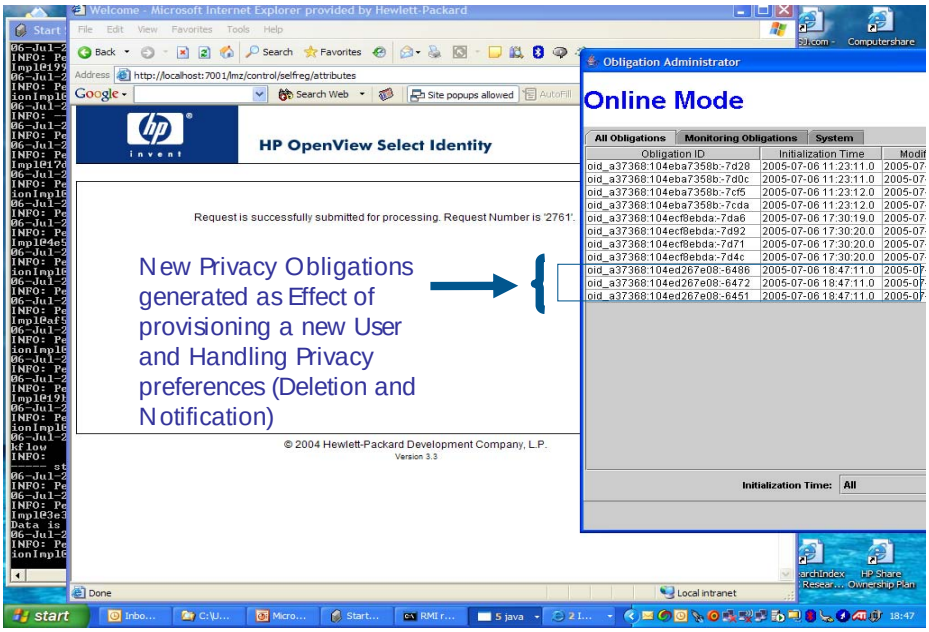


Figure 41: Creation of new Privacy Obligations

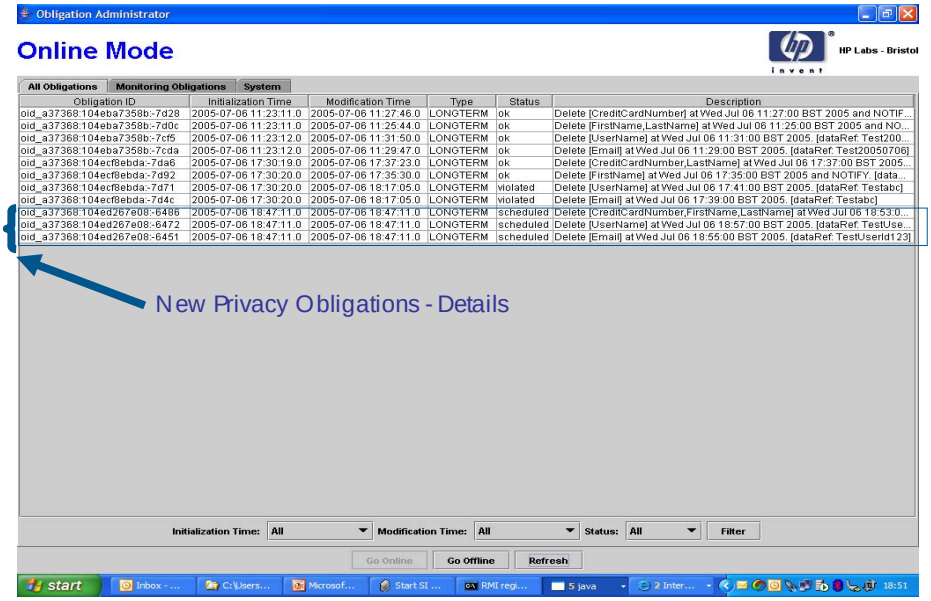


Figure 42: Details about New Privacy Obligations

Figure 43 shows the instant where one of the newly created obligations is triggered for enforcement (based on time criteria). The enforcement of this obligation will trigger HP Select

Identity's workflow mechanisms to handle user information (in this specific case to delete data).

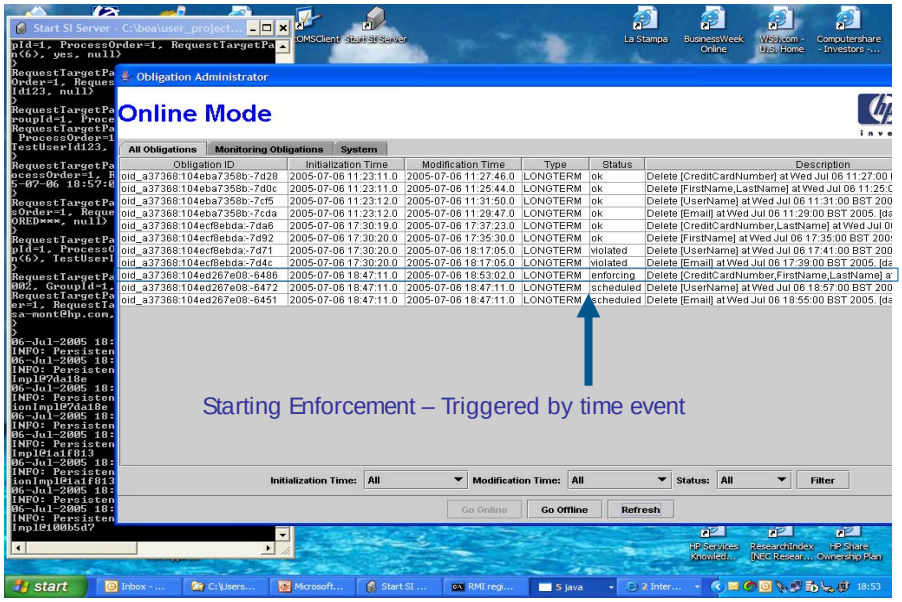


Figure 43: Starting the Enforcement of a Privacy Obligation

Figure 44 shows the instant when the privacy obligation has been successfully enforced. The obligation management system shows the result of this process.

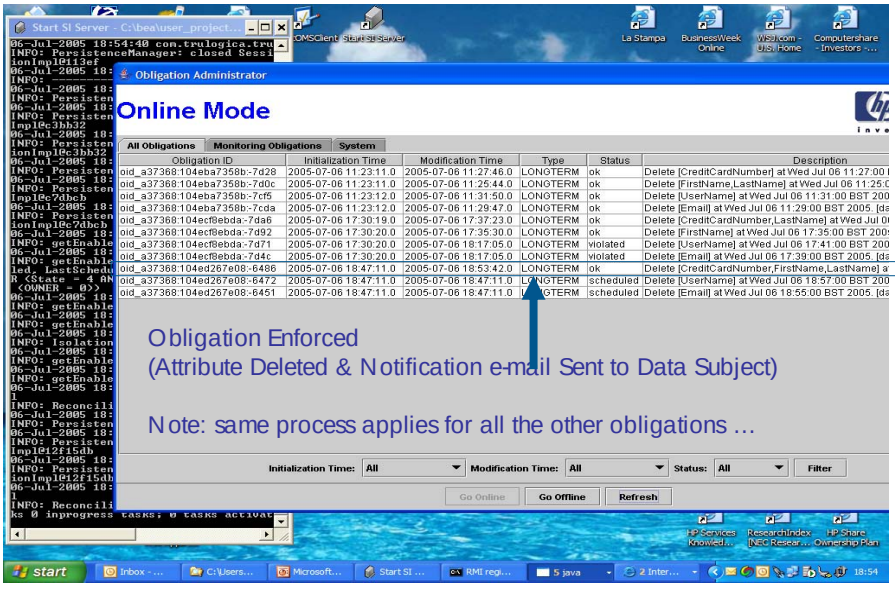


Figure 44: Privacy Obligation Enforced

Once the privacy obligation has been enforced, the system starts monitoring its status (for a predefined period of time, for compliance reasons): it will report violations or divergences from the expected status of the targeted data.

10 Discussion

Chapter 4 described a few important requirements to be addressed by an obligation management framework:

- **Explicit modeling of privacy obligations;**
- **Association of obligations to data;**
- **Mapping obligations into actions;**
- **Monitoring obligations;**
- **Accountability management;**
- **Tracking the evolutions of obligation policies;**
- **Compliance of refined obligations to high-level policies;**
- **Complexity and cost of instrumenting applications and services.**
- **Dealing with long-term obligation aspects;**
- **Integration with current identity management solutions;**
- **User involvement and awareness.**

This chapter compares these requirements against the functionalities provided by our system and highlights aspects that have not yet been fully addressed and open issues.

The obligation management framework (and related implemented system) described in this thesis mainly targets the modeling of privacy obligations, the enforcement and monitoring of these obligations.

Our representation of privacy obligations explicitly describes which data obligations refer to, which events trigger obligations and which actions need to be executed. In this context, monitoring obligations means verifying that the outcome of enforcing these actions is preserved.

Issues could arise in terms of managing the association of privacy obligations to personal data in case these data are not really static but subject to frequent changes of their storage locations and disclosures to third party. The next section of this chapter describes in more details this problem and potential approaches to address it.

We assume that enterprises are willing to be compliant with privacy policies and, more specifically, privacy obligations. However, because of its nature, the system described in this paper has to be considered as a trusted system. It must be deployed by keeping in mind good security practices, especially for the platforms that will host our system modules. These modules are critical hence they require to be secured accordingly. Additional trust and accountability can be added by hardening the audit server and involving trusted third parties in the monitoring of the enforcement of obligation policies.

In particular, the audit server is fundamental to log all the activities and management decisions made during the processing, enforcement and monitoring of privacy obligations. The implementation of the audit server and how to harden it are beyond the scope of this work. More details about possible approaches can be found in [BaSh04], [Bald04].

In terms of tracking the evolution of privacy obligations, at the moment our system centralises the storage of privacy obligations along with their management. It can potentially support the management of versions of privacy obligations over time and enable the tracking of their changes (and related applicability contexts) for auditing and accountability reasons. Our cur-

rent work does not cover this aspect: additional work has to be done in this space, in particular to enhance the authoring and administration tools (and related UIs).

Dealing with the problem of assuring compliance of "operational obligations" to high-level obligation policies is not trivial. It involves authoring and refinement tools that track the policy refinement process and ensure that any change in the refinement chain is properly propagated. This issue is common to all policy management frameworks. It has not been explicitly addressed by this work. Preliminary work in this space is described in [CaBG99].

The approach described in our architecture is almost transparent to the data affected by privacy obligations. However, applications and services might require some instrumentation, especially if applications/service-based events need to be detected and managed in order to trigger privacy obligations. We are currently investigating how an Event API (to be used by applications and services) and a related event management framework can be used to accommodate different needs and requirements.

Our system explicitly focuses on the management and enforcement of obligations: this does not imply that it has to happen independently by other privacy aspects, such as permissions. It should be considered as a sub-system of a more comprehensive privacy management framework and integrated with identity management solutions. This aspect is currently researched at HP Labs by the author.

All the system modules can be distributed to avoid potential bottlenecks and central points of failure, without compromising the overall security and integrity of the system.

When dealing with long-term privacy obligations it is also important to ensure the reliability and longevity of the platforms running our system components and the survivability of the involved data and obligations. Work has already been done in this space, including [Ande96], [EFL+98], [KBC+00], [Neum99], [WBS+00], and can be leveraged.

Our prototype has demonstrated the feasibility of integrating our obligation management system with a state-of-the art, identity management system.

In this context user involvement is of primary importance. Users (data subjects) can specify their privacy preferences that will be automatically turned into privacy obligations to be fulfilled by enterprises.

Additional work has to be done on the administrative UI of our system, in order to allow users to be involved also in the lifecycle management of these privacy obligations.

10.1 Open Issues

Managing privacy obligations on "static" data stored within enterprise data repository is relatively easy. Issues arise when the overall environment is dynamic and data can be moved around: in this case the association of data to obligations policies can be broken or be left in an inconsistent state.

To address this important issue we are currently exploring a variant of the architecture shown in Figure 17, where stronger mechanisms are introduced to manage the association of obligations to data. In this context we talk about "sticky" obligations.

In our approach, personal data is encrypted in "data envelopes" and strongly associated to privacy obligations by using cryptographic techniques. A key management system is introduced to deal with this task as a subsystem of the Obligation Server [Casa04a].

Figure 45 shows a variant of our system architecture to encompass these capabilities.

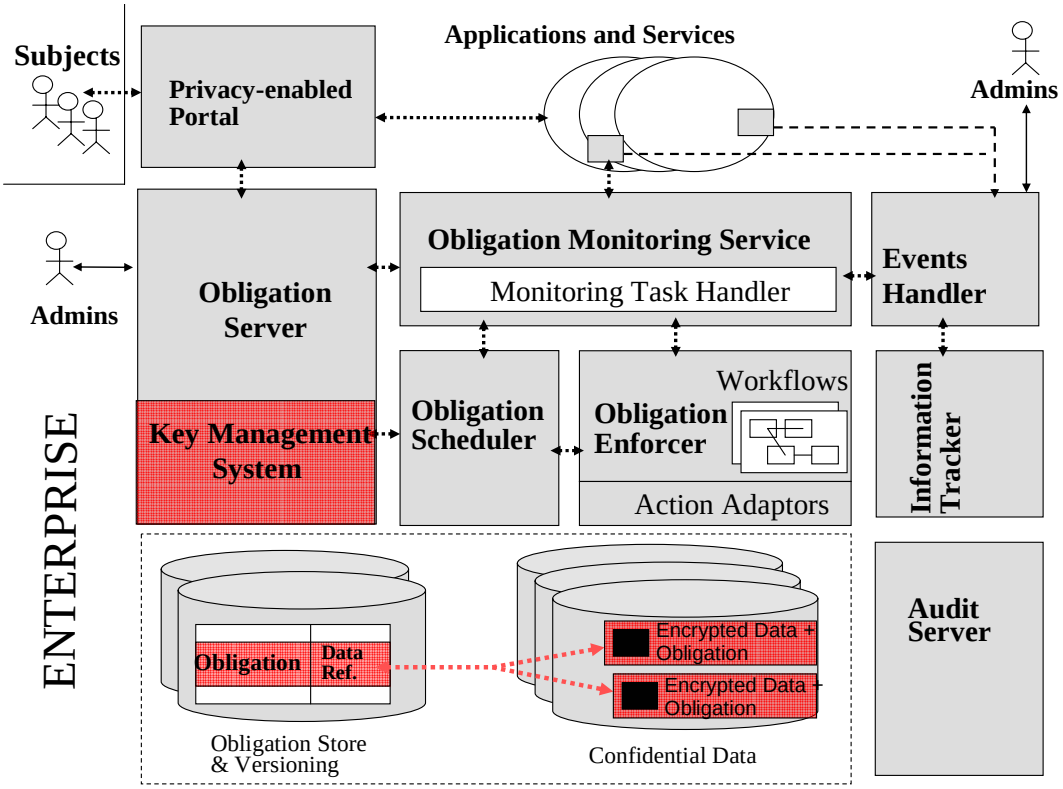


Figure 45: Extended Architecture

Specifically, data envelops are encrypted with a public key [HFPS99] associated to the key management system. An alternative (but conceptually similar) approach is also feasible by leveraging the Identifier-based Encryption (IBE) schema [BoFr01], [Cock01].

The triple consisting of <*obligation policy, encrypted envelope, obfuscated data*> is stored as a replacement of the original data.

The obligation policy will contain a reference to the competent Obligation Server but it can omit the reference to confidential data, as the policy is now directly associated to this data. In this way, the encrypted confidential data can be moved around and transmitted to other parties without an upfront control.

The receiving party has to interact with the Obligation Server to decrypt the data: this allows the system to track and audit where the data are, check for relevant obligations and update its obligation store. Of course, once data have been disclosed, they can be misused. The auditing process is required to create more accountability and to identify responsibilities. Additional research must be done to understand how to limit (unnecessary) disclosures of data by locally performing operations (when this is possible), under the authorization of data subjects.

The basic principles and additional details on how the above approach can be implemented and accountability assured are described in [CaPB03], [Casa04a], [Casa04b]. This approach requires additional research and investigations, in terms of its feasibility, in terms of performance and impact on existing infrastructure.

Additional research is also required to better understand the impact - on applications and services - of enforcing privacy obligations that require the encryption of personal data.

This process is transparent to our obligation management system as our system does not directly interact with applications and services: it simply "transforms" personal data as an effect of enforcing obligations. However, the consequences and implications of changing the status of personal data (from clear data to encrypted data) need to be properly investigated in the IT framework where our system will be deployed. At the moment this is an open issue that requires further work.

Another important issue that has not been fully addressed by the current work is the management of potential conflicts between privacy obligations managed by the system.

Conflicts could arise at the time a new privacy obligation is added to the system or existing obligations are modified.

An initial investigation has been done in this space to build an additional system module that identifies sets of obligations "targeting" the same personal data (by analysing the Target element of privacy obligations) and checks, within each set, for potential conflicts.

We envisage an approach that consists of analysing and comparing the content of obligations' events and actions, for all the privacy obligations in a given set, to spot inconsistencies.

For example, actions defined by two different privacy obligations (targeting the same data) might dictate contradictory actions, at different points of time (e.g. two different triggering dates), such as deleting user' personal data at one date and afterwards notifying the user by using such data (that is now deleted). In this case the second obligation cannot be enforced as no user's data is available anymore.

At the moment we are exploring different types of conflicts that could occur along with strategies for identifying and highlighting them to privacy administrators. More work has to be done in this space.

Finally, further work has to be done to investigate the overhead on the IT infrastructure generated by the event management framework used by our obligation management system. This is still an open issue: we need to understand how flexible and usable the event management framework provided by the PRIME project is and if and how we can leverage any of the existing event management frameworks, available in commercial solutions.

10.2 Future Research Topics and Directions

The core architectural design and the basic functionalities of our obligation management system will remain the same. Our future plans in this area are around two specific types of activities:

1. Extending some of the current functionalities of the obligation management system;
2. Adding new functionalities to the obligation management system.

In terms of extending some of the current Obligation Management functionalities, we plan to:

- Extend the set of supported events within privacy obligations, to include more general fine-grained access control-based events and events related to trust and contextual aspects of the system;

- Fully explore the implications of using an external event management framework. This includes integrating the current OMS prototype with the PRIME event management framework and/or a commercial one;
- Extend the set of implemented actions within privacy obligations, to include complex workflows, involving also human interactions;
- Explore additional aspects of privacy obligations, such as handling exceptions and dealing with the NOT operator. Extend the privacy obligations accordingly;
- Explore the implications of dealing with large sets of privacy obligations and how to address related performance aspects, in particular in terms of the set of events that must be managed. This might include handling "parametric" privacy obligations that can be used for different data records (and users) and mechanisms to automatically cluster sets of obligations that shares the same triggering events (in order to minimise the set of managed events);
- Extend the current Adminstrator UI to provide a more fine-grained management of privacy obligations, including a complete lifecycle management process of privacy obligations (by dealing with the modification and deletion of privacy obligations, in addition to adding them), that could be accessed by both administrator and end users;
- Extend the OMS prototype integrated with HP Select Identity, as an effect of our better understanding of its deployment in a real-world context, consisting of heterogeneous provisioned systems;
- Standardize the obligation policy format, possibly in a W3C standardization body (yet to be defined): this will probably happen in the process of standardizing the various types of "policies/rules" managed in the EU PRIME project.

In terms of adding new functionalities, we plan to extend the obligation management system by:

- Dealing with obligation policy conflicts, by researching and building new mechanisms that address this problem;
- Dealing with sticky obligations i.e. obligations strongly associated to personal data both when these data are stored by an enterprise and transmitted between parties;
- Researching and providing "reputation management" feedback functionalities to further involve data subjects in the process of evaluating how well organisations/enterprises are compliant with the enforcement of their promises, including privacy obligations derived by data subjects' privacy preferences. This functionality will be provided by additional modules within the obligation management system and external modules to be deployed at the data subject site.

Related to the last point - providing reputation management feedback functionalities - the obligation management system currently only provides basic feedback mechanisms to end-users, for example via periodic notifications, based on obligations defined by the end-users.

This functionality really depends on the correct setting of preferences and obligations done by end-users during the execution of "external" processes, such as a self-registration phase or the disclosure of personal data. It does not handle service-side obligations.

End-users have little control of their obligations and how they are currently managed. In the current system, an end-user cannot easily interact with the obligation management system at the enterprise side and get a complete view of how their personal data have been handled, based on specified obligations and preferences.

Our plan is to leverage and extend the current obligation management system by:

- Researching and developing a new module, called *Obligations Status and Compliance Manager* (OSCM), to provide end-users' with a focused overview of the fulfilment of privacy obligations defined on their PII data.

  This includes sub-modules to report the current status of specific obligations defined on end-user's PII data (both by end-users and service-side administrators), logging information and highlighting activities done on these PII data as a consequence of enforcing obligations, listing all the notifications and interactions that happened in a past with the end-user (in a predefined timeframe) and reporting any incident or problem occurred during the enforcement of these obligations.

  In particular, a new UI is going to be provided so that end-users can check and retrieve all this information in a simpler and more focused way than the current UI (that is designed mainly for administrators' needs). These new functionalities are complementary to the basic notification functionality already available in the obligation management system:

- Extension of the obligation management system to represent and handle the required events and information that the OSCM component might need;

- Extension of the current obligation format to support additional feedback mechanisms to end-users.

  Research has to be done to identify any additional "notification" mechanisms that might be relevant to provide feedback to end-users, in addition to the current notification system based on e-mails. This might include researching and building enhanced workflow capabilities for the obligation enforcer;

- Extension of obligation management system to handle additional "enterprise-side" obligations, i.e. obligations defined by enterprise administrators (as an effect of laws and guidelines) that have an impact on data subjects' personal data.

  The OSCM components will provide feedback and compliance information also on these service-side obligations.

It is important to notice that the OSCM component will provide data subjects' with the "view" of how the enterprise has been "good" at managing data subjects' obligations and preferences on their personal data.

The information gathered via the OSCM component has still to be checked and compared by end-users against their personal experience of what actually happened.

This will be done at the client-side, via an extension of the current work done at HP Labs, Bristol in the context of the PRIME project: the outcome of this comparison will contribute to shape the reputation of enterprises.

Most of the related research and development work stated in this chapter are going to be carried on in the context of the PRIME project.

11 Conclusions

The management of privacy obligations is important for enterprises to preserve their reputation and brand, be compliant with legislation and customers' requirements and increase business opportunities. This thesis analyses the concept of privacy obligation and describes important issues and requirements that need to be kept into account by enterprises when dealing with privacy obligations.

In our vision privacy obligations need to be considered as independent, first-class entities, not subordinated to access control aspects and explicitly managed within an obligation management framework.

We introduce an obligation management model and a technical solution to deal with the explicit management of privacy obligations including transactional/short-term, long-term and ongoing privacy obligations. We describe our obligation management system to deal with the monitoring, enforcement, and tracking of privacy obligations.

To demonstrate the feasibility of our approach and how it can be deployed in the real world, a fully working prototype has been implemented.

This prototype has also been integrated with the user provisioning and self-registration capabilities of a state-of-the-art identity management solution, HP Select Identity, to show how our work can effectively enable privacy-oriented data lifecycle management in enterprises.

Current open issues include the problem of strongly associating privacy obligations to confidential data in dynamic environment, dealing with accountability management, dealing with policy conflicts and dealing with a user-friendly lifecycle management and administration of privacy obligations.

Our research and work on these aspects is in progress. Most of this work will be done both in the context of the EU PRIME project and as HP Labs, to increase the functionality of our prototype integrated with HP Select Identity.

Acknowledgements

I would like to thank Pete Bramhall, HP Labs, for his comments, inputs and review of this thesis.

Thanks to Kwok-Nga (Annie) Chan for her contribution to implement aspects of the obligation management prototypes [kwok04], in the context of the EU PRIME project and its integration with HP Select Identity.

Thanks to Jan De Clercq, for our discussions and interactions in the space of identity management and for providing the content of Figure 6.

Thanks to HP Software Business Unit (SGBU) for sharing with us their HP Select Identity product and solution, their support during our integration and the documentation they provided to us (Figure 34 has been retrieved from related documentation).

Thanks also to HP Labs for authorising me to describe and discuss in this thesis my R&D work on privacy obligation management done as a researcher at HP Labs.

References

[AKSX02] Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: "Hippocratic Databases", IBM Almaden Research Center, 2002

[Ande96] Anderson, R. J.: "The Eternity Service", Proc. PRAGO-CRYPT 96, CTU Publishing House, Prague, 1996

[Ande05] Anderson, A.: "A Comparison of EPAL and XACML (v2.1) ", Sun Microsystems, http://research.sun.com/projects/xacml/CompareEPALandXACML.html, 2005

[Arno00] Arnold, T.: "Internet Identity Theft: A Tragedy for Victims", White Paper, SIIA, 2000

[BaFS03] Baldwin, A., Ferreira, A., Shiu, S.: "Towards Accountability for Electronic Patient Records", 16th IEEE Symposium on Computer-Based Medical Systems, CBMS 2003, June 2003

[Bald04] Baldwin, A.: "Enhanced accountability for electronic processes", 2nd international conference on trust management. Lecture notes in computer science, vol. 2995, Springer, 2004

[BaSh04] Baldwin, A., Shiu, S.: "Enabling shared audit data", IJIS, 2004

[BBC+03] Beres, Y., Bramhall, P., Casassa Mont, M., Gittler, M., Pearson, S.: "On the Importance of Accountability and Enforceability of Enterprise Privacy Languages", position paper, W3C Workshop on the long-term future of Enterprise privacy Languages, http://www.w3.org/2003/p3p-ws/pp/hp1.pdf, 2003

[BJSW02] Bettini, C., Jajodia, S., Sean Wang, X., Wijesekera, D.: "Obligation Monitoring in Policy Management", 2002

[Blum02] Blum, D.: "Toward Federated Identity Management", Burton Group, 2002

[BoFr01] Boneh, D., Franklin, M.: "Identity-based Encryption from the Weil Pairing", Crypto 2001, 2001

[Burt02] Burton Group: "User Authentication", Burton Group, 2002

[CaBG99] Casassa Mont, M., Baldwin, A., Goh, C.: "POWER prototype: Towards Integrated Policy-based Management", HPL Technical Report, HPL-1999-126, 1999

[CaBP03] Casassa Mont, Bramhall, P., Pato, J.: "On Adaptive Identity Management: The Next Generation of Identity Management Technologies", HP Labs Technical Report, HPL-2003-149, 2003

[CADT00] Coates, D., Adams, J., Dattilo, G., Turner, M.: "Identity Theft and the Internet", Colorado University, 2000

[CaPB03] Casassa Mont, M., Pearson, S., Bramhall, P.: “Towards Accountable Management of Privacy and Identity Information”, ESORICS 2003, 2003

[Casa04a] Casassa Mont, M.: "Dealing with Privacy Obligations: Important Aspects and Technical Approaches", TrustBus 2004, 2004

[Casa04b] Casassa Mont, M.: "Dealing with Privacy Obligations in Enterprises", ISSE 2004, 2004

[CaTB05] Casassa Mont, M., Thyne, R., Bramhall, P.: "Privacy Enforcement with HP Select Access for Regulatory Compliance", HP Labs Technical Report, HPL-2005-10, 2005

[CBG+02] Casassa Mont, M, Bramhall, P., Gittler, M., Pato, J., Rees, O.: "Identity Management: a key e-business enabler", HP Labs Technical Report, HPL-2002-164, presented at SSGRR2002s, L'Aquila, Italy, 2002

[CHM+02] Chen, L., Harrison,K., Moss, A., Soldera, D., Smart,N.P.: "Certification of Public Keys within an Identity Based System", Proc. 5th Int. Information Security Conference (ISC), 2002, LNCS 2433, Springer-Verlag, 2002

[Cock01] Cocks, C.: "An Identity Based Encryption Scheme based on Quadratic Residues", Communications - Electronics Security Group (CESG), UK. http://www.cesg.gov.uk/site/ast/idpkc/media/ciren.pdf, 2001

[Copp00] COPPA: "The Children's Online Privacy Protection Act", http://www.ftc.gov/bcp/conline/pubs/buspubs/coppa.htm, 2000

[DDLS01] Damianou, N., Dulay, N., Lupu, E., Sloman, M.: "The Ponder Policy Specification Language", 2001

[Decl02] De Clercq, J.: "Single Sign-On Architectures", proceedings pp. 40-58, InfraSec 2002, Bristol, UK, 2002

[EFL+98] Ellison, R.J., Fisher, D.A., Linger, R.C., Lipson, H.F., Longstaff, T.A., Mead, N.R.: "Survivability: Protecting your Critical Systems", Proceeding of the International Conference of Requirements Engineering, 1998

[Epal04] IBM: "The Enterprise Privacy Authorization Language (EPAL) ", EPAL 1.2 specification. http://www.zurich.ibm.com/security/enterprise-privacy/epal/, IBM, 2004

[EuCo05] European Commission: "Data Protection in the European Union", http://europa.eu.int/comm/justice_home/fsj/privacy/index_en.htm, 2005

[FeKu92] Ferraiolo, D., Kuhn, R.: "Role-based Access Control", NIST, 1992

[GaBl02] Gamby, R., Blum, D.: "Developing Identity Management and Directory Services Architecture Principles", Technical Positions and Templates, The Burton Group, 2002

[Gaw01] Gaw, J.: "Digital Identity Solutions: A Road Map for Software and Services", IDC, 2001

[GLB03] GLB: "Gramm-Leach-Bliley Act: Privacy of Consumer Financial Information", http://www.ftc.gov/privacy/glbact/glboutline.htm, 2003

[Hewl05a] Hewlett-Packard (HP): "HP OpenView Select Identity: Overview and Features", http://www.openview.hp.com/products/slctid/index.html, 2005

[Hewl05b] Hewlett-Packard (HP): "Trusted Systems Lab", Bristol, UK, http://www.hpl.hp.com/, 2005

[HFPS99] Housley, R. Ford, W., Polk, W., Solo, D.: "RFC2459: Internet X.509 Public key Infrastructure Certificate and CRL Profile", IETF, 1999

[Hipa05] HIPAA: "Health Insurance Portability & Accountability Act (HIPAA) - Administrative Simplification Laws - Overview", http://www.dhs.state.mn.us/main/groups/business_partners/documents/pub/dhs_id_016236.hcsp, 2005

[Ibmt04] IBM Tivoli: "IBM Tivoli Storage Manager for Data Retention", 2004

[KaSc02] Karjoth, G., Schunter, M.: "A Privacy Policy Model for Enterprises", IBM Research, Zurich. 15th IEEE Computer Foundations Workshop, 2002

[KaSW02a] Karjoth, G., Schunter, M., Waidner, M.: "Platform for Enterprise Privacy Practices: Privacy-enabled Management of Customer Data", 2nd Workshop on Privacy Enhancing Technologies, Lecture Notes in Computer Science, Springer Verlang, 2002

[KaSW02b] Karjoth, G., Schunter, M., Waidner, M.: "Privacy-enabled Services for Enterprises", IBM Zurich Research Laboratory, TrustBus 2002, 2002

[KBC+00] Kubiatowicz, J., Bibdel, D., Chen, Y., Czerwinski, S., Eaton, P., Geels D., Gummadi, R., Rhea, D., Weatherspoon, H., Weimer, W., Wells, C., Zao, B.: "OceanStore: An Architecture for Global Scale Persistent Storage", University of California, Berkeley, ASPLOS 2000, 2000

[Kwok04] Chan, K.: "Obligation Management System: A Component of Identity Management", MsC, SDIA, Newcastle University, 2004

[Laur04] Laurant, C.: "Privacy International: Privacy and Human Rights 2003: an International Survey of Privacy Laws and Developments, Electronic Privacy Information Center (EPIC)", Privacy International. http://www.privacyinternational.org/survey/phr2004/, 2004

[LiAP05a] Liberty Alliance Project: "Liberty Project web Site", http://www.projectliberty.org/, 2005

[LiAP05b] Liberty Alliance Project: "Liberty Architecture Overview", http://www.projectliberty.org/, 2005

[Micro05] Microsoft Corporation: "Next Generation Secure Computing Base", http://www.microsoft.com/resources/ngscb/default.mspx, 2005

[Neue02] Neuenschwander, M.: "Meta-directory Services and the Emerging Enterprise Data Network", The Burton Group, 2002

[Neum99] Neumann, P.G.: "Practical Architectures for Survivable Systems and Networks", SRI International, Army Research Lab, 1999

[Netf05] NetForensics: "NetForensics Web site", http://www.netforensics.com/, 2005

[OASI05a] OASIS: "eXtensible Access Control Markup Language TC", XACML 2.0, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml, 2005

[OASI05b] OASIS: "OASIS Security Services TC: SAML", v. 2.0, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security, 2005

[Oecd80] OECD: "OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data", http://www1.oecd.org/publications/e-book/9302011E.PDF, 1980

[Pato03] Pato, J.: "Identity Management: Setting the Context", HP Labs Technical Report, HPL-2003-72, 2003

[PaVi01] Parr, B., Villars, R.: "Digital Identities: The Coming Struggle for the Future of the net", IDC, 2001

[Pear02] Pearson, S. (ed.): "Trusted Computing Platforms", Prentice Hall, 2002

[Penn02a] Penn, J.: "IT Trend 2002: Directories and Directory-Enabled Applications", IdeaByte, 2002

[Penn02b] Penn, J.: "Market overview: user Account Provisioning", GIGA Information Group, 2002

[Prim05] PRIME Project: "Privacy and Identity Management for Europe, European RTD Integrated Project under the FP6/IST Programme", http://www.prime-project.eu.org/, 2005

[Priv04] Online Privacy Alliance: "Guidelines for Online Privacy Policies", http://www.privacyalliance.org/, Online Privacy Alliance, 2004

[PrLa05] Privacy Law: "News and Information on The Law of Privacy", http://www.privacylaw.net/, 2005

[RGC+05] Roessler, T., Hogben, G., Casassa Mont, M., Pearson, S.: "Position Paper: Rule Language Requirements for Privacy-Enabled Identity Management", W3C Workshop, Rule Languages for Interoperability 2005, http://www.w3.org/2004/12/rules-ws/paper/59/, 2005

[Safe05] Safe Harbour: "Safe Harbour Overview", http://www.export.gov/safeharbor/sh_overview.html, 2005

[Senf03] Senf, D.: "Identity Management in the Enterprise: Consolidating eBusiness Interactions", IDC, 2003

[Sens05] SenSage: "SenSage Web site", http://www.sensage.com/, 2005

[ScAs02] Schunter, M., Ashley, P.: "The Platform for Enterprise Privacy Practices", IBM Zurich Research Laboratory, 2002

[Smit01] Smith, R.E.: "Authentication: From Passwords to Public keys", Addison-Wesley, 2001

[SOX05] SOX: "Sarbanes-Oxley Act", http://www.sarbanes-oxley.com/, 2005

[Syno05] Synomos: "Synomos Align 3.0", http://www.synomos.com/, 2005

[TCPA01] Trusted Computing Platform Alliance: "TCPA Main Specification", Version 1.1, http://www.trustedcomputing.org, 2001

[Volc01] Volchkov, A.: "Revisiting Single Sign-on. A Pragmatic Approach in a New Context", pp. 39-45, IT Pro, IEEE, 2001

[W3C01] W3C: "XML Key Management Specification (XKMS)", http://www.w3.org/TR/xkms/, 2001

[W3C02] W3C: "The Platform for Privacy Preferences 1.1", http://www.w3.org/P3P/, 2002

[W3C03a] W3C: "XML Signature WG", http://www.w3.org/Signature/, 2003

[W3C03b] W3C: "XML Encryption WG", http://www.w3.org/Encryption/, 2003

[W3C03c] W3C: "Simple Object Access Protocol (SOAP)" v.1.2, http://www.w3.org/TR/SOAP/, 2003

[W3C03d] W3C: “Extensible Markup Language (XML)”, http://www.w3.org/XML/, 2003

[W3C04] W3C: “Resource Description Framework (RDF)”, http://www.w3.org/RDF/, 2004

[WBS+00] Wylie, J.J., Bigrigg, M. W., Strunk, J. D., Ganger, G. R., Kiliccote, H., Khosia, P.K.: "Survivable Information Storage Systems", IEEE Computer, 2000