Privacy Policy Enforcement in Enterprises with Identity Management Solutions

Marco Casassa Mont, Robert Thyne[1]
Trusted Systems Laboratory
HP Laboratories Bristol
HPL-2006-72
April 25, 2006*

privacy, privacy management, policy enforcement, privacy-aware access control, automation, identity management

People are usually asked by enterprises and other organizations to disclose their personal information to access web services and engage in business interactions. Enterprises need this information to enable their business processes. This is unlikely to change, at least in the foreseeable future. When collecting personal data, enterprises must satisfy privacy laws and policies along with addressing people's expectations on how their data should be handled. Currently much is done by means of manual processes, in particular in terms of privacy enforcement: these processes are prone to mistakes and hard to comply. Automation can help enterprises to deal with these privacy management issues, in particular the enforcement of privacy policies on collected personal data. Enterprises have already been investing in identity management solutions: they require that approaches to automate privacy management should keep into account and leverage these solutions. This paper discusses our research and development work to automate the enforcement of privacy policies in enterprises. Our model of privacy policy enforcement is introduced along with the technical details of a related prototype, integrated (as a proof of concept) with HP Select Access, a state-of-the-art identity management solution. This technology is currently under productisation. We discuss our current results and next steps.

* Internal Accession Date Only
[1]Hewlett-Packard, Software Business Organisation, Toronto, Canada

Approved for External Publication

Privacy Policy Enforcement in Enterprises with Identity Management Solutions

Marco Casassa Mont, Robert Thyne, *Hewlett-Packard*

Abstract— **People are usually asked by enterprises and other organizations to disclose their personal information to access web services and engage in business interactions. Enterprises need this information to enable their business processes. This is unlikely to change, at least in the foreseeable future.**

When collecting personal data, enterprises must satisfy privacy laws and policies along with addressing people's expectations on how their data should be handled. Currently much is done by means of manual processes, in particular in terms of privacy enforcement: these processes are prone to mistakes and hard to comply. Automation can help enterprises to deal with these privacy management issues, in particular the enforcement of privacy policies on collected personal data. Enterprises have already been investing in identity management solutions: they require that approaches to automate privacy management should keep into account and leverage these solutions.

This paper discusses our research and development work to automate the enforcement of privacy policies in enterprises. Our model of privacy policy enforcement is introduced along with the technical details of a related prototype, integrated (as a proof of concept) with HP Select Access, a state-of-the-art identity management solution. This technology is currently under productisation. We discuss our current results and next steps.

***Index Terms*— Privacy, Policy Enforcement, Privacy-aware Access Control, Automation**

I. Introduction

People usually must disclose their personal data to enterprises to access services on the Internet and engage in business transactions. Enterprises and other organizations need this information to enable their business processes and interactions.

By collecting, storing, processing and potentially disclosing personal data, enterprises are subject to privacy laws and related legislation. Privacy laws [1,2] and privacy guidelines, such as OECD [3], dictate that enterprises should clearly state the purposes for which they are collecting personal data and should take into account the consent given by people to use their data for these purposes, along with any additional privacy preference. In addition, personal data should be deleted once its retention is not required anymore. Openness and transparency over how data is processed, manipulated and disclosed to third parties are also key requirements. People should be notified of changes affecting the management of their personal data and they should retain a degree of control over it. Compliance to all these aspects must be monitored and any violations promptly reported and addressed. Large enterprises that are geographically distributed across different nations might need to comply with different privacy laws.

Privacy management is therefore important for enterprises: it has implications on their compliance with regulations, their reputation, brand and customers' satisfaction [19,20].

Privacy policies can be used to represent privacy laws and guidelines: they describe people (data subjects)'s *rights* on their personal data, *permissions* given to enterprises and *obligations* that enterprises need to fulfil when handling personal data.

We focus on those enterprises (actually the vast majority) that have realized how important is to correctly deal with privacy policies and people's preferences: they are looking for appropriate and reasonable solutions to achieve this.

Enterprises have already been investing in identity management solutions to automate the management of personal and identity information. This includes [16]: (1) solutions to store identity information and credentials: (2) solutions to use this information (among other things) for access control and authorization; (3) single-sign-on mechanisms and federated identity management solutions to simplify and enable multi-party interactions; (4) provisioning and user account management solutions to simplify users' self-registration process and provision users' information to various enterprises' systems and data repositories.

However, in terms of privacy management, much is still done by means of manual processes or by ad-hoc solutions, which make them difficult and expensive to comply. Enterprises require a simplification of the involved processes and better control. Automation of privacy management is a viable approach to address (part of) these aspects.

Most of the technical work currently done in this space focuses on auditing and reporting solutions that analyse logged events and check them against privacy policies. This addresses compliance aspects of privacy management. However, also operational aspects of privacy need to be addressed. In particular, the automation of the enforcement of privacy policies is very important to guarantee that personal

Manuscript received March 31, 2006.

M. Casassa Mont is with Hewlett-Packard Laboratories, Trusted System Lab, Bristol UK (e-mail: marco.casassa-mont@ hp.com).

R. Thyne is with Hewlett-Packard, HP Software Business Organisation, Toronto Canada (e-mail: robert.thyne@hp.com).

data is accessed, used, disclosed and managed according to these policies.

This paper describes our systemic approach to automate the enforcement of privacy policies. We describe our research and technology and how it can be integrated with enterprise middleware solutions, in particular identity management solutions.

II. ADDRESSED PROBLEM

This paper focuses on the problem of how to automate the enforcement of privacy policies on personal data stored within enterprises by keeping into account privacy laws, enterprise guidelines and people's privacy preferences.

Privacy policies describe privacy *rights*, *permissions* and *obligations*. In this paper we specifically address the problem of enforcing privacy policies on personal data at the access time, by keeping into account – among other things – related *rights* and *permissions* (inclusive of dealing with stated purposes and users' consent). Our approach to deal with the enforcement of privacy *obligations* is discussed in [20,21]: it is complementary to the work discussed here and is beyond the scope of this paper.

At the very base, the enforcement of privacy policies on personal data is about privacy-aware access control. Our goal is to address this by developing a privacy enforcement framework and a systemic approach that can be leveraged by current enterprises' identity management solutions.

III. IMPORTANT ISSUES AND REQUIREMENTS

In this paper, we will use the terms "data subjects", "people" and "users" in an interchangeable way. We consider scenarios where users are asked by enterprises (e.g. service providers) or other organisations to disclose their personal information in order to access services, engage in transactions or access information.

We want to enable users to specify their privacy preferences on how their data should be managed, ask for their explicit consent and allow them to customise constraints about the usage of their data. We want to provide users with degrees of control on their personal data.

We also want to enable enterprises to: keep into account users' privacy preferences and enforce them; explicitly author privacy policies, deploy and enforce them during accesses, manipulations and transmission of personal data.

Enterprises currently need tools to achieve this and, at the same time, allow them to leverage their identity management solutions.

The (technological) enforcement of privacy permissions and rights (on stored personal data) requires extended access control and authorization mechanisms that check privacy permissions against data requestors' credentials, check the consistency of data requestors' *Intent* (to access personal data) against stated purposes and take into account the consent given by data subjects [19]. Enterprise services or applications that need to access and manipulate personal data for various reasons should be subject to the enforcement of these privacy policies.

Traditional access control systems are necessary but not sufficient to enforce privacy policies on personal data. They are mainly based on "access control lists" and enforcement mechanisms that keep into account only the identities of data requestors, their rights and permissions and the types of actions that are allowed/disallowed on the involved resources. These systems do not keep into account additional aspects relevant to privacy: the stated purposes for collecting data and data subjects' consent - i.e. properties usually associated to collected data - the *Intent* of data requestors and any additional enterprise or customized data subjects' constraints - see Figure 1.

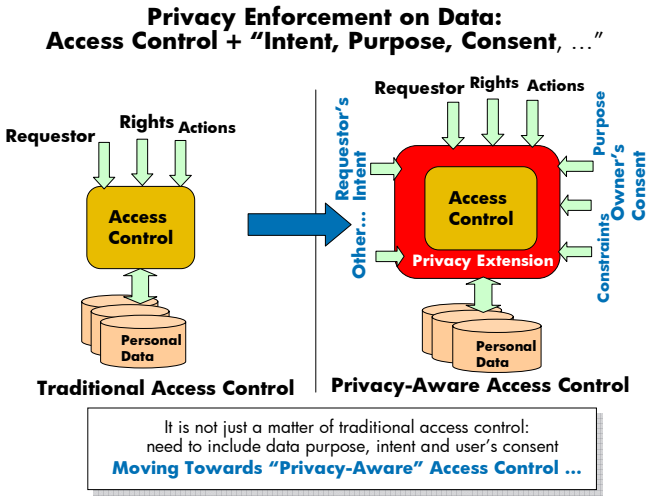

Fig. 1. Moving Towards Privacy-aware Access Control.

To address the above issues and move towards privacy-aware access control, it is important to satisfy the following core requirements:

1) Explicit modeling of personal data stored by enterprises;

2) Explicit definition, authoring and lifecycle management of privacy policies;

3) Explicit deployment and enforcement of privacy policies;

4) Integration with traditional access control and identity management systems;

5) Simplicity of usage of all the involved system;

6) Support for auditing.

A more comprehensive analysis and discussion of these aspects can be found in [19,24].

Section IV of this paper presents our work in this space. Sections V and VI present related work and a discussion of the results we achieved with our work.

IV. OUR WORK

This section describes our work to automate the enforcement of privacy policies on personal data stored by enterprises. Our approach consists of researching and developing solutions that can be leveraged by current enterprise identity management solutions.

We focus on the following (typical) enterprise identity management process, which occurs when a new user wants to access services or applications that might require personal data to deal with related business transactions:

1. The user (data subject) is asked to access a self-registration web site and provide their personal information and other requested data. Some privacy preferences might also be asked to the user and stored. The user later on will be allowed to change their information and preferences;
2. Provisioning and user account management solutions are used (for example [23]) by the enterprise to manipulate user's information and store (parts of) it within relevant data storages. The same provisioning solutions will take care of creating user accounts across enterprise' involved systems and set proper access control on these resources. These provisioning solutions will track changes happening on stored information and ensure that information is kept aligned and consistent;
3. As an effect of the previous steps, authorization and access control systems might be provisioned (by means of access control constraints, new user accounts, etc.) and will be able to grant (or deny) access to services.

The above process usually focuses only on the automation of identity management and security aspects. Privacy aspects are either not included or their enforcement is not automated. In addition, personal data is stored in enterprise data repositories subject only to the enforcement of security aspects.

Figure 2 summarises how we aim at automating privacy policy enforcement.

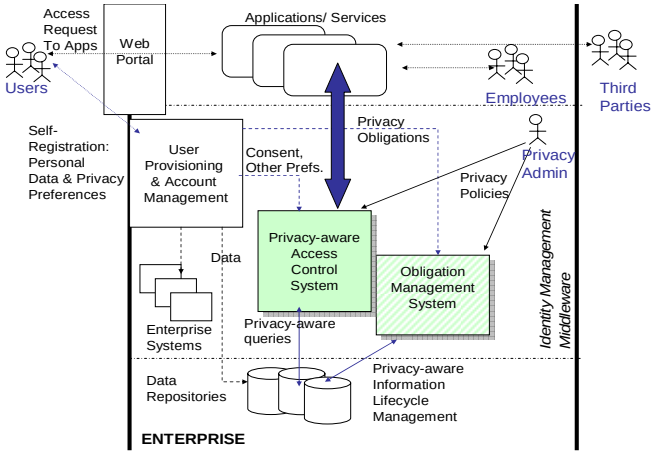

Fig. 2. Privacy Enforcement with Identity Management Solutions

Specifically our work wants to:

1. Enable users to explicitly define privacy preferences (inclusive of consent and data retention) and customise them during and after their self-registration phase;
2. Use users' privacy preferences, during the provisioning phase, to:
    a. Configure *extended* access control systems to provide privacy-aware access to personal data: this includes ensuring that these systems can keep track of stated purposes, data subjects' consent and other privacy constraints;
    b. Turn parts of these privacy preferences (such as retention date, notification choices, etc.) into explicit privacy obligations to be enforced by enterprises.
3. Allow enterprises to author, deploy and enforce "enterprise-side" privacy policies and privacy obligations derived from privacy laws and internal guidelines.

As anticipated, it is beyond the scope of this paper to cover our work on the management and enforcement of privacy obligations. See [20, 21] for more details. The remaining part of this paper will focus on our work to manage and enforce privacy policies on stored personal data – at the access time. A preliminary description of this work has already been published in [19, 24].

Our approach is based on a privacy-aware access control model. This model extends traditional access control models (based on users/groups, users' credentials and rights, access control lists and related policies) by explicitly dealing with the stated purposes for which data is collected, checking - at the access request time - the *Intent* of requestors against these purposes, dealing with data subjects' consent and enforcing additional access conditions and constraints on defined by data subjects and/or enterprise administrators [1,2,3]. See Figure 3.

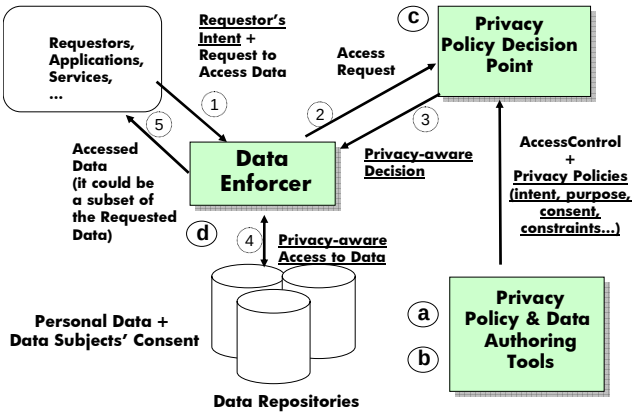

Fig. 3. Model of our Privacy-aware Access Control System.

Work has already been done in the space of privacy-aware access control (see section V). However, our aim is to develop technology that can leverage state-of-the art Identity Management solutions and be integrated with existing enterprises' access control solutions. This is currently an open issue that we want to address.

Figure 3 shows the main aspects of our model:

a) **A mechanism for the explicit modelling of personal data, subject to privacy policies**: this mechanism provides a model/description of the personal data subject to privacy policies, including the type of the data

repository (database, LDAP directory, etc.), its location, the schema of these data, types of attributes, etc.;

b) **An integrated mechanism for authoring privacy policies along with traditional access control policies**: it is a Policy Authoring Point (PAP) to allow privacy administrators to describe and author privacy policy constraints and conditions (including how to check consent and data purpose against requestors' *Intent* and how to deal with data filtering and transformation, etc.) along with more traditional access control policies based on security criteria (such as who can access which resource, given their rights and permissions);

c) **An integrated authorization framework for deploying both access control and privacy-based policies and making related access decisions**: it is a privacy-aware Policy Decision Point (PDP);

d) **A run-time mechanism - referred here as the "Data Enforcer" - for intercepting attempts to access personal data and enforcing decisions based on privacy policies and contextual information**, e.g., *Intent* of requestors, their roles and identities, etc. It is a privacy-aware Policy Enforcement Point (PEP). This mechanism is in charge (among other things) of dealing with the transformation of queries to access personal data (e.g. SQL queries) and filtering part of the requested data, if their access is not authorised for privacy reasons.

The Data Enforcer component plays a key role to enforce privacy policies on stored personal data. At "run-time", attempts to access personal data are intercepted and managed in the following way - see Figure 3:

1. **A request from a data requestor to access personal data is intercepted by the Data Enforcer**. Available information about the requestor (credentials, identity, etc.) is collected, along with their *Intent* (that can be explicitly passed as a parameter or could be predefined in the application/service making the request);
2. **The Data Enforcer interacts with the privacy policy decision point** by passing information about the request (including the ***Intent***) and the requestor;
3. **The privacy policy decision point makes a decision, based on available privacy policies and the context** (request, requestor's information, etc.). This decision is sent back to the Data Enforcer. It can be any of the following types:
    a. ***Deny***: access to data is denied;
    b. ***Deny + conditions***: access to data is denied. Some conditions are sent back to the requestors. The satisfaction of these conditions (for example passing the *Intent* or authenticating) could change the outcome of the decision;
    c. ***Allow*****:** access to data is granted;
    d. ***Allow + conditions****:* access to (part of the) data is allowed, under the satisfaction of the attached conditions. Among other things, these conditions might require data filtering, data transformations and manipulations.
4. The Data Enforcer enforces this decision. In particular, if the decision is ***"Allow + conditions"*** the Data Enforcer might have to manipulate the query (query pre-processing) and/or transform the requested personal data (result post-processing), before returning the result to the data requestor;
5. Data (or alternatively no data) is returned to the data requestor, based on the enforced decision.

Figure 4 shows a simple example based on this model where an attempt to access personal data is made by an enterprise employee.

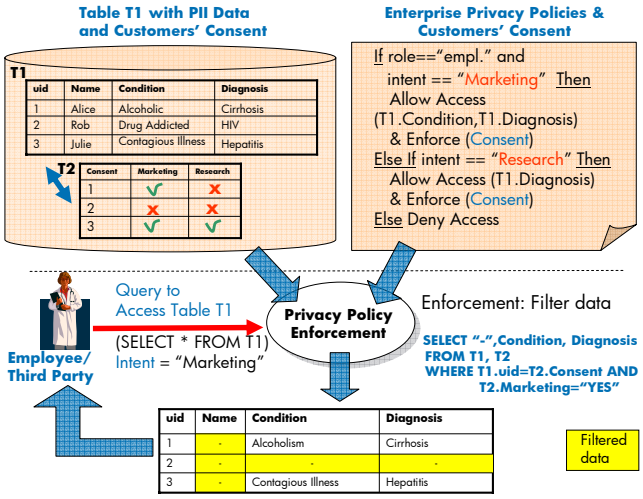

Fig. 4. Example of Privacy Policy Enforcement.

In this example, the employee's declared *Intent* (i.e. *Marketing*) is consistent with the stated purposes for collecting data (*Marketing, Research*) – declared in the associated privacy policy. However the employee is trying to access – via a SQL query - more data than she is allowed to. The SQL query is intercepted by the enforcement point (Data Enforcer) and transformed on-the-fly (before being submitted to the database) in a way to include constraints based on data subjects' consent and the filtering of data. The transformed query is then submitted to the database. In this example privacy is achieved by pre-processing and transforming the query before actually interacting with the database.

We implemented our privacy enforcement model in a prototype by leveraging and extending HP Select Access. HP Select Access [14] is a leading-edge access control solution. It provides policy authoring, policy decision and policy enforcement capabilities via the following components:

- **Policy Builder**: it is a graphical tool to author access control policies (PAP) on resources managed by the system;
- **Validator**: it is a Policy Decision Point (PDP). It makes access control decisions based on the access control policies (authored with the Policy Builder) and contextual information, such as the identity of a requestor;
- **Web Enforcer plug-in**: it is a Policy Enforcement Point

(PEP) for web resources.

The current commercial version of HP Select Access does not handle ***data*** as managed resources: it only deals with traditional access control policies on ***web resources*** i.e. web pages, servlets, etc. exposed in a web server. Additional functionalities have been added to HP Select Access, in our prototype, to explicitly deal with privacy-aware access control on personal data, as shown in Figure 5.

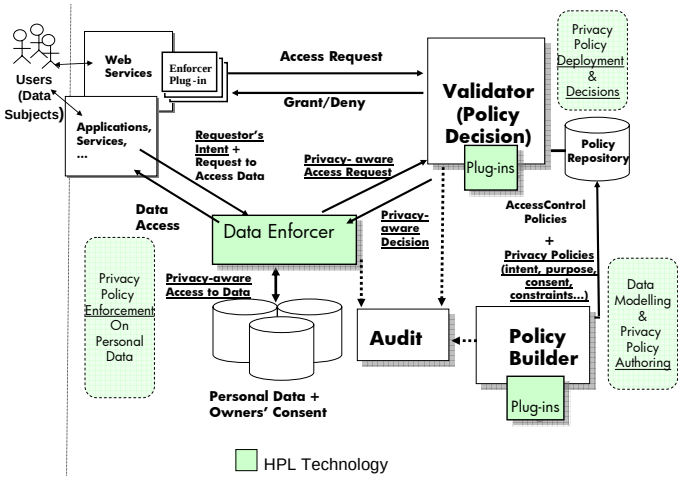

Fig. 5. Extended HP Select Access to deal with Privacy Policy Enforcement

The specific extensions are:

- **The Policy Builder has been extended to represent "data resources"** (databases, LDAP directories, virtual-directories, their schemas, etc.) in addition to traditional IT resources (such as web resources);
- **The Policy Builder has been extended to graphically author privacy policies on "data resources" in addition to traditional access control policies**: a set of privacy-related plug-ins has been implemented to allow checking (at the enforcement time) the requestor's *Intent* against the stated data storage purposes, take into account data subjects' consent and data retention policies and describe how the accessed personal data must be filtered, obfuscated or manipulated, etc.;
- **The Validator has been extended to make privacy-aware decisions.** Plug-ins, correspondent to the ones used in the Policy Builder, have been implemented. This enhanced-version of the Validator can now make ***"Allow + conditions"*** decisions as described in our model;
- **A Data Enforcer has been built and added to the framework:** this is a new functionality added to HP Select Access. It is in charge of enforcing privacy decisions made by the Validator, as previously described in our model. The Data Enforcer proxies managed data repositories (e.g. databases, LDAP directories, virtual directories, etc.): we envisage that a family of Data Enforcers (sharing a common logic but differentiated by add-ons dealing with different types of data resources) need to be built, because of the different semantic of different data repositories. As a proof of concept, we implemented a Data Enforcer as a JDBC proxy for RDBMS databases.

The above functionalities address and satisfy the core requirements described in section III for privacy enforcement on personal data.

The next sections provide more details on our technical solution and the prototype that we have implemented. We introduce a healthcare scenario and describe a simple example of personal data that must be protected by the enterprise. Based on this example, we provide more technical details about two key components of our work: (1) Privacy Policy Authoring Component and (2) Data Enforcer. We also describe the implications for applications and services that need to access this personal data.

A. HealthCare Scenario: Example of Personal Data

We consider a healthcare scenario where personal data has to be disclosed by patients to a Healthcare Service Provider (enterprise) to get medical diagnosis and advices. This service provider uses our technology – i.e. our Privacy Policy Enforcement Solution, integrated to HP Select Access - not only to secure the access to its web services but also to protect accesses to personal data, once it has been stored within its data repositories. Patients can specify their privacy preferences in terms of consent (usage of data for stated purposes) and data retention. Our solution ensures that these privacy preferences and additional privacy constraints are satisfied – at the access time.

In this context, we consider a simple example of personal data stored within an enterprise RDBMS database, called ***PatientDB.*** This database contains two key tables:

1. ***PatientRecords***: this table contains patients' personal data, including: *Name, DateofBirth, Gender, SSN (i.e. Social Security Number), Address, Location, e-mail, Lifestyle Notes, GP, HealthSituationNotes, Consultations, Hospitalisations and FamilyHistory*;
2. ***PrivacyPreferences***: this table contains patients' privacy preferences, including: *Name, MarketingPreference, ResearchPreference, Third Party Disclosure, RegistrationDate and DataRetention Period*.

At the self-registration time a new patient can provide his personal data and subsequently update it. This information is stored in the *PatientRecords* table. At the same time the patient can express his/her personal preferences, stored in the *PrivacyPreferences* table, in terms of:

1. **Which Purposes** their data can be used for. In this example three purposes are stated by the enterprise: *Marketing, Research* and *Disclosure to Third Parties*. The patient can allow or deny the use of their data for each of these purposes. Personal data must not be accessed if the requestor's *Intent* is inconsistent with these preferences;
2. **Retention Time**: this defines the period of time by which personal data can be retained by the enterprise.

Data must not be accessed after the expiration time and it must be deleted.

In this example, a patient's record is linked to his/her related privacy preferences record by using the *Name* attribute, as a unique identifier associated by the enterprise to each patient.

In a real-world scenario, multiple tables could have been used to store both personal data (and other data) and privacy preferences, across one or more data repositories (not necessarily relational databases). Privacy preferences on personal data could have been more fine-grained, focusing at the attribute level instead of being at the record level.

The database structure described in this example has been kept deliberately simple for illustration purposes. Of course, our technology can cope with the complexity of real-world scenarios, thanks to the flexibility of our privacy policy authoring and privacy policy enforcement solutions.

B. *Privacy Policy Authoring*

In the current HP Select Access solution [14], traditional access control policies can be graphically authored via a ***<Resources x Users>*** matrix within the ***Policy Builder*** (see Figure 6. Resources are represented in the vertical, left tree-panel; Users are represented in the horizontal, top tree-panel). Both Resources (i.e. web resources) and Users can be structured in hierarchies to better reflect enterprise needs [14].

In this matrix, the *intersection* of a User (or set of Users) with a Resource (or set of Resources) can be used by an administrator to set an *Allow* or a *Deny access right* for that User on that Resource.

Alternatively, an administrator can define more refined access control criteria, by specifying a Policy (Rule). This is achieved by using the ***Rule Editor*** tool that allows administrators to graphically author policies and associate them to a User(s) to control their access to a Resource(s). See Figure 6. Policy constraints are described via graphical components. The Rule Editor has a set of predefined, embedded constraints that can be used when authoring policies: new ones can be added by developing plug-ins.

In the current HP Select Access solution, access control policies can include constraints dictating stronger authentication, impose time-based access (e.g. 9:00AM-5:00PM), etc. Our work has leveraged and extended the HP Policy Builder component in order to *model personal data* and *author related privacy policies*.

"Personal Data" has been modeled as an additional type of Resource within the Policy Builder (in the Resource panel) – see Figure 6.

The model of personal data is based on describing the type of data repository used to store this data (e.g. RDBMS, LDAP, Meta-directory, Virtual-directory, etc.), the actual data repository name and details about the data schema.

In the context of our healthcare scenario and our example of personal data, we use a RDBMS database. The data resources (i.e. a patient data table in the database) involved in this examplecan be modeled as follows: "***Data Resources/Databases/PatientDB/Tables/***". Under this hierarchy we positioned the ***PatientRecords*** table, containing the actual data to be protected – i.e. subject to privacy policies.

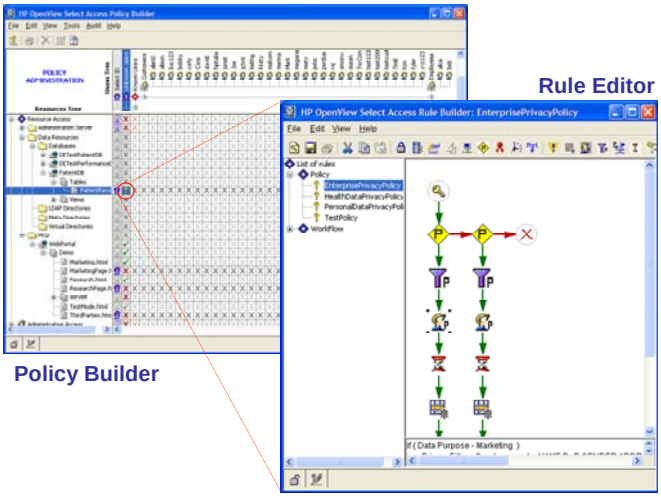

Fig. 6. Extended Policy Builder to Author Privacy Policies on Stored Data

Figure 6 also illustrates a simple privacy policy associated to this ***PatientRecords*** table – as it has been authored in the *Rule Editor*. Figure 7 provides additional details.

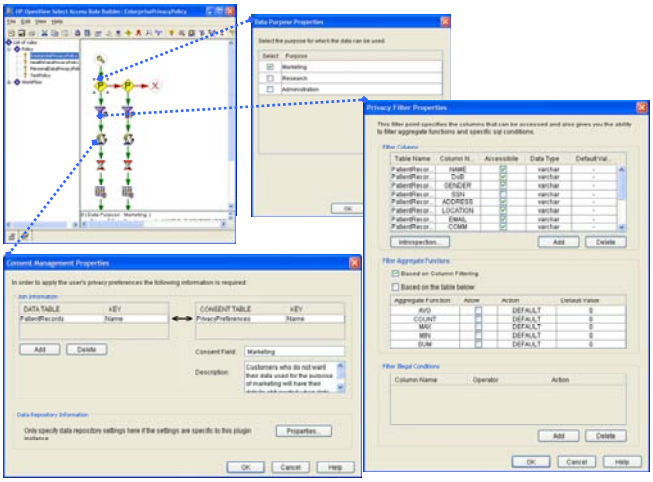

Fig. 7. Key privacy plug-ins: Purpose Checking, Privacy Filtering, Consent Management and Data Expiration

Thanks to our extension, also privacy policies can be graphically authored in the *Rule Editor*: they resemble a flowchart as they describe core checking points and actions that need to be done at the enforcement time. Specifically this includes:

- Checking a data requestor's *Intent* (to access data) against stated purposes. In the example described in Figure 6 (and in more details in Figure 7), personal data can be accessed only for two stated purposes: *Marketing* and *Research*. A requestor's *Intent* must be matching one of these purposes. Any other type of *Intent* will prevent the requestor from accessing any data. This is achieved via our specific **purpose**

management plug-in, added to the Rule Editor (Figure 7 – yellow rhomboidal component);

- For each purpose (i.e. *Marketing*, *Research* in the current example) - allowing degrees of access to data. Each branch of the privacy policy, related to a specific data purpose, is used to describe constraints in terms of ***data filtering***, ***consent management*** and ***data expiration***. This is achieved via privacy-aware plug-ins of the Rule Editor that we have built. Figure 7 shows an example of the properties of some of these plug-ins.

Additional information about these plug-ins follow:

A) The **data filtering plug-in** can be used by a policy administrator to:

- List a set of protected (personal data) attributes and describe which one should be filtered at the access time i.e. replaced with a default value. For example, in Figure 7 (case where *Intent = "Marketing"*) a list of all the fields in the ***PatientRecords*** table is displayed. In this example, a few fields (i.e. *SSN, GP, HeathSituationNotes, Consultations, Hospitalisations and Family History)* need to be filtered and replaced with a default value (in the example the "-" string). This is important to prevent a requestor to access the value of these attributes but still allow access to other attributes that are not subject to constraints;
- List a set of (system or user-defined) *functions/stored procedures* that might access personal data and describe how to handle them. In our example we consider SQL aggregation functions (e.g. *MIN*, *MAX, AVG*, etc.). Our plug-in allows the administrator to describe if the result provided by these functions should be filtered (i.e. replaced with default values) or if the SQL query should fail because of their presence. This is important to preserve privacy: information could be indirectly retrieved or deduced about people by using these functions/stored procedures;
- List a set of SQL conditions and describe how to handle them. In our example we consider SQL conditions - in the WHERE or HAVING clauses - such as "***DateOfBirth>DateXYZ"***. Dealing with this aspect is also important to preserve privacy: personal information could be indirectly retrieved or deduced about people by using well crafted conditions.

Additional constraints can be added to this filtering plug-in based on the specific need and type of data.

B) The **consent management plug-in** is used to specify how to retrieve the consent information specified by a user (and associated to his/her personal data) and how the system should check it at the access time, given the data requestor's *Intent*. Personal data must not be accessed if no consent is given. Figure 7 shows an example of how the plug-in is used to link personal data stored in the ***PatientRecords*** table to preferences stored in the P***rivacyPreferences*** table. Specifically this is achieved by using the *Name* attribute (patient's unique ID) as a key to link records in the two tables. This allow the administrator to specify which privacy preference should be analysed at the enforcement time and allow the privacy enforcement system to decide about a user's consent. In the example under analysis (where the data requestor's *Intent* matches the *Marketing* purpose) the *MarketingPreference* attribute in the P***rivacyPreferences*** table contains the relevant patient's preference i.e. his/her consent (*Yes/No* flag) to access their data for *Marketing* reasons.

In general more complex links can be expressed by using multiple associations, should fine-grained preferences be expressed not only at the record level but also at the attribute level.

C) The **data expiration plug-in** is conceptually very similar to the consent management plug-in. It is used to specify how to retrieve the "expiration date" information associated to each user's personal data and allows the systems to check it at runtime against the current time. Personal data must not be accessed if it "has expired". In our example the *Data Retention Period* attribute within the P***rivacyPreferences*** table contains this expiration date.

Once a privacy policy has been authored and associated to a set of Data Resources and Users, it is locally stored within the HP Select Access system. It will be used at the access time by the Validator component (Policy Decision Point) to make a privacy-aware decision and provide related directives to the Data Enforcer.

C. Data Enforcer

Our solution allows the enforcement of privacy policies on personal data stored in different data repositories, including RDBMS databases, LDAP directories, meta/virtual-directories.

In this section we provides additional technical details about the Data Enforcer that we have developed in our prototype to intercept SQL queries in RDBMS databases and enforce privacy policies on requested data. The internal architecture of our Data Enforcer is shown in Figure 8.

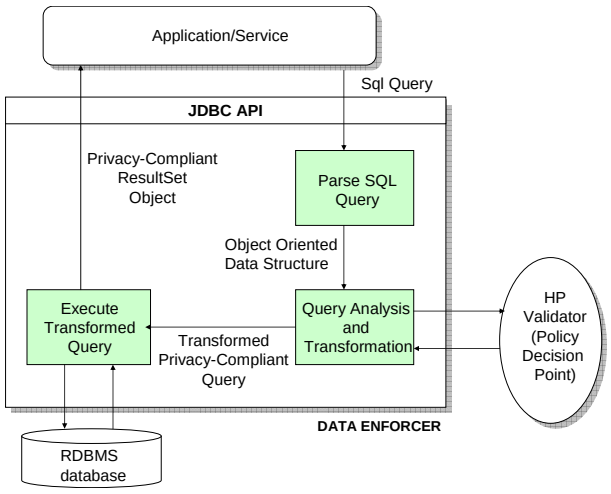

Fig. 8. Internal Architecture of the Data Enforcer

This Data Enforcer is based on a JDBC Proxy (i.e. JDBC

driver extended with interception mechanisms). Applications and services basically do not need to be modified apart from having to use this JDBC driver. More details on other implications are provided in the next section.

We assume that applications/services use standard JDBC APIs to access data via SQL queries. The Data Enforcer intercepts these SQL queries and processes them.

The *Intent* of a data requestor (e.g. user, application or service) could be implicit in its role: this ensures the most transparent interaction between applications and our Data Enforcer. In case the *Intent* information has to be explicitly passed by the application to the Data Enforcer (because it depends on business logic and/or current interaction), we support two mechanisms to achieve this: (1) the *Intent* information can be added by the application at the end of its SQL query – before submitting it; (2) the *Intent* information can be passed as a property object by the application, via the JDBC API - *getConnection* method.

In our healthcare scenario, we consider –as an example – the case where web services/applications (accessed by patients, employees and third parties) explicitly handle the *Intent* information and add this information at the end of their SQL queries. For example:

SELECT * FROM PatientRecords; #PrivacyContext: INTENT=Marketing

The "**Parse SQL Query**" component of the Data Enforcer - see Figure 8 - intercepts incoming SQL queries (SELECT, CREATE, UPDATE, INSERT, etc.), parses them and generates an explicit tree-based (object-oriented) representation of these queries. This representation clearly identifies, given an arbitrary SQL query, what the involved *data resources* are (e.g. DB tables, fields, etc.), *Intent* information, SQL conditions on data, etc. In our example, the ***PatientRecords*** in ***PatientDB*** is a data resource that needs to be protected.

The "**Query Analysis and Transformation**" component – see Figure 8 - checks with the Validator if any privacy policy applies, for each involved data resource. In doing this it will pass relevant contextual information (requestor's identity, requestor's *Intent*, etc.) to the Validator. If privacy policies apply, related decisions are recorded. They might include the filtering of some of the data associated to specific fields, the fact that consent has to be enforced, etc.

In our example (described in the *Privacy Policy Authoring* section), if the data requestor tries to access personal data in ***PatientRecords*** with a Marketing *Intent*, it will be subject to a privacy policy saying that a few fields (*SSN, GP, HeathSituationNotes, Consultations, Hospitalisations and Family History)* must be filtered and replaced with a default value (in the example the "-" string). In addition the policy was requiring to enforce patients' consent and data expiration. This means that, for example, the previous simple SQL query (***SELECT * FROM PatientRecords***) will be automatically transformed by the Data Enforcer into:

SELECT
PatientRecords.Name, DateofBirth, Gender, '-' AS SSN, Address, Location, E-mail, Conn, LifestyleNotes, '-' AS GP, '-' AS HealthSituationNotes '-' AS Consultations, '-' AS Hospitalization, '-' AS FamilyHistory
FROM *PatientRecords, Privacy Preferences*
WHERE
PatientRecords.Name = PrivacyPreferences.Name AND PrivacyPreferences.MarketingPreference = 'Yes' AND PrivacyPreferences.DataRetentionPeriod<now

Please notice that this transformation keeps into account the constraints previously described in our example of privacy policy, defining:

- Fields to be filtered out, along with their default values (e.g. *'-' AS SSN);*
- How to keep into account personal data (patients' records) and their preferences (i.e. … ***.FROM*** *PatientRecords, Privacy Preferences);*
- How to enforce a patient's consent to use/not use their data for Marketing purposes (i.e. … ***WHERE*** *PatientRecords.Name = PrivacyPreferences.Name AND PrivacyPreferences.MarketingPreference = 'Yes' AND* …);
- How to enforce the expiration date for each patient's record (i.e. **WHERE** … *AND PrivacyPreferences.DataRetentionPeriod<now*).

This transformation of SQL queries happens on-the-fly: in general, if specific fields need to be filtered out (because a privacy policy says so), these fields are replaced in the query representation with default values (as described in the policies). If data subjects' consent has to be enforced, additional JOIN conditions are added into the query representation to check for data subjects' consent information.

The *Query Transformation* process can involve:

- **SELECT statements**: based on relevant privacy policies, functions and stored procedures could be filtered (and replaced with default values) or their presence could cause the query to fail;
- **CREATE/ALTER VIEW statements**: the content of these statements are processed and exactly the same transformation process apply, to any SELECT query embedded in these statements;
- **UPDATE/INSERT statements**: as above, for example if any SELECT statement is embedded.

The ***"Query Analysis and Transformation"*** component – see Figure 8 - also parses and interprets any allowed combination of the above statements, including nested statements (e.g. nested SELECT queries), UNION of queries, etc.

This component is aware of SQL variants, in terms of syntax and used keywords, as defined by the SQL engines of *Oracle DB* and *Microsoft SQL Server*.

The outcome of this module is a transformed SQL query

that keeps into account all stated privacy constraints and is still compatible with the original SQL query (requested by the application).

This query is sent to the "**Execute Transformed Query**" - see Figure 8 - that sends it the RDBMS system so that it is executed by the real SQL RDBMS engine. The result of this privacy-compliant query is directly sent back to the application/service.

It is important to notice that the above types of transformations only require the *pre-processing* of SQL queries. No *post-processing* of query results is involved. This is a very efficient process: the RDBMS will further optimize the transformed SQL query and return the expected result. Some initial encouraging performance results are presented in Section VI.

Our Data Enforcer could also post-process query results, in those cases where this is required and return them to the requestor: however, at the moment, we have found no need to proceed in this direction, given the kind of privacy policies we want to enforce.

D. Implications for Applications and Services

As anticipated in the previous section, the overall impact for enterprise applications and services is meant to be minimal. Nevertheless applications and services that want to access protected data need to be slightly modified as follow:

1. They have to interact with the Data Enforcer in order to ensure that privacy policies are enforced. In case of RDBMS databases, our JDBC Proxy Driver must be used. For example, a Java application, at the time to connect to a database has to declare that it will use our Data Enforcer (e.g. *Class.forName ("org.hplpivacy.jdbc.jdbcDriver"); String url = "jdbc:hplprivacy:microsoft:sqlserver://localhost:3241; DatabaseName='PatientDB'"*) and use it in Connection method calls (e.g. *Connection con = DriverManager. getConnection(url, user_name,pwd)*);
2. If required, they have to pass the *Intent* of the data requestor, by attaching it to the end of the SQL statement, or passing it as a property, in the JDBC API (getConnection method). Of course if this information is omitted (i.e. there is no declaration of *Intent*) the query is likely to return no data, in those cases where the relevant privacy policy checks the *Intent* against stated purposes.

V. RELATED WORK

A common approach adopted by enterprises to enforce privacy-aware access control policies on personal data (when this is not just a matter of good practices and manual processes) consists of hard-coding them within applications and services or building ad-hoc solutions. This approach is suitable for very simple and static environments: it shows all its limitations and maintenance costs in case of complex and dynamic organizations that need to adapt to changes. As described in the requirements section, to explicitly address the automation problem, a model of the relevant personal data is required. Privacy policies need to be authored, deployed, enforced and audited. This requires the definition of a comprehensive privacy-aware access control model and systems that implement it.

Relevant work in this direction, for privacy management and enforcement in enterprises is described in [4,5,6,7]. An Enterprise Privacy Architecture is introduced and described in [7]. This approach is further refined and described in the Enterprise Privacy Authorization Language (EPAL) specification [8]. However these papers only provide general guidelines and do not describe a deployable solution within current identity management solutions.

Important related work on actual privacy enforcement on personal data has been done on Hippocratic databases [9] and similarly on Oracle Databases (Virtual Private Database [25]). The drawback of this approach is that it mainly focuses at the database level, specifically on RDBMS data repository architectures and related data schemas. The enforcement of privacy policies might need to span across a broad variety of data repositories and legacy systems to include LDAP directories, meta and virtual directories, file systems and legacy systems. It might need to incorporate higher-level views and perspectives than just the database-level perspective. Our work supports this.

Other relevant work in the space of privacy policy enforcement on RDBMS databases is discussed in [15]. This work describes the concept of purpose-based access control and its application to RDBMS databases by labeling data with stated purposes within data tables. Based on this labeling process, the system only returns the data that can be accessed for given purposes. This approach is complementary to our approach. Our approach does not label data and operates at a different level of abstraction: allowed purposes are defined in our privacy policies and these policies linked to relevant personal data. We will explore if we can exploit this approach by integrating it with ours – in particular in case fine-grained (at the attribute level) consent and purpose management is required.

In terms of commercially available solutions, IBM Tivoli Privacy Manager [10, 11] provides mechanisms for defining fine-grained privacy policies and associating them to data. On one hand this solution provides the required privacy enforcement functionalities. On the other hand this approach dictates strong constraints on how applications need to be developed and how personal data has to be stored and administered: it might require some duplications of administrative and enforcement frameworks (e.g. it might require the parallel usage of Tivoli Access Manager) and it is vertically-based on other IBM products and solutions.

Other products, such as HP Select Federation [12] and ePok [13], focus on single-sign-on and related privacy aspects: they enforce privacy rules on personal data in federated environments where data is disclosed by an organization (or an identity provider) to other parties.

Our work specifically addresses the problem of enforcing

privacy policies on personal data stored in a broad variety of data repositories within enterprises. Personal data can be accessed by different types of requestors, including people, applications and services. It includes related aspects of modeling managed data and authoring privacy policies.

Our work aims at not being invasive for applications and services: privacy policies are managed in an explicit way, in conjunction with traditional access control policies and not hard-coded in applications and services. Our work avoids duplication of effort by providing a single, integrated framework for authoring, administering and enforcing both traditional access control and privacy policies. This has been demonstrated in our work that leverage and extended HP Select Access [14] also to enforce privacy policies on personal data.

Our work is also consistent with work done in the context of the PRIME Project [22], in terms of the privacy policies and preferences that need to be defined and enforced on personal data. We are involved in this project and we will ensure compliance of our approach and solutions to the outcomes of this project.

VI. Discussion and Next Steps

The aim of our work is not really to deal with hostile enterprise environments: we provide mechanisms and solutions to enterprises willing to enforce privacy policies on personal data, within their identity management solutions.

A critical aspect of our proposed solution is that users, applications or services might explicitly need to declare their *Intent* to access data.

Having the freedom to explicitly state the *Intent* to access data might be necessary in some contexts: for example a doctor that urgently needs to access data to cure a patient in danger of life, could state a specific *Intent* to bypass some patient's privacy preferences and access this data.

However, in general, this mechanism could potentially be abused to access personal data that actually should not to.

Because of this, it is very important for enterprises to audit data requestors' attempts to access personal data, their credentials, their stated *Intent* and related privacy decisions. We believe that this is a viable way to ensure accountability [18]. It can be achieved by using commercial auditing solutions, such as [17], to audit our privacy policy enforcement system.

The prototype we have developed to enforce privacy policies with HP Select Access is a proof of concept. However it shows the feasibility of our work to address the enforcement of privacy policies in a systemic way, integrated with state-of-the-art identity management solutions. This has been achieved by using the same policy authoring tools and policy decision/enforcement framework both for security-based access control policies and privacy policies. This simplifies the overall management process for enterprise administrators and avoids duplication of efforts (otherwise needed in case when multiple tools/solutions need to be used).

We are refining and extending this technology for its productisation: to achieve this we are collaborating with the HP Software Business organisation.

However, it is important to highlight the fact that our model and technology are general purpose: they can be leveraged, integrated and deployed in other identity management contexts, beyond HP identity management solutions.

At the moment the enforcement of privacy policies with HP Select Access mainly consists in enforcing data subjects' consent, constraints on data purposes and data expirations via data filtering.

Current performance tests and analysis (done on databases of sizes ranging from 100K to 500K records) are promising. Figure 9 provides a summary of performance tests we carried on with a MS SQL 2000 database, in the context of our healthcare scenario and by storing personal data and preferences in the ***PatientRecords*** and ***PrivacyPreferences*** tables.

The graph in Figure 9 shows that the time required to return all records in a "SELECT * FROM PatientRecords" query grows linearly with the database size (number of records).

Actually, the time required to return all records in case of a privacy-transformed SQL query is less than the time required to executed the original (un-modified) SQL query: this because records might be dropped when no users' consent was given (for a given purpose). Noticeably, this time is also less than the time required to return all the records by directly asking the database the transformed SQL query, without the interception of our Data Enforcer.

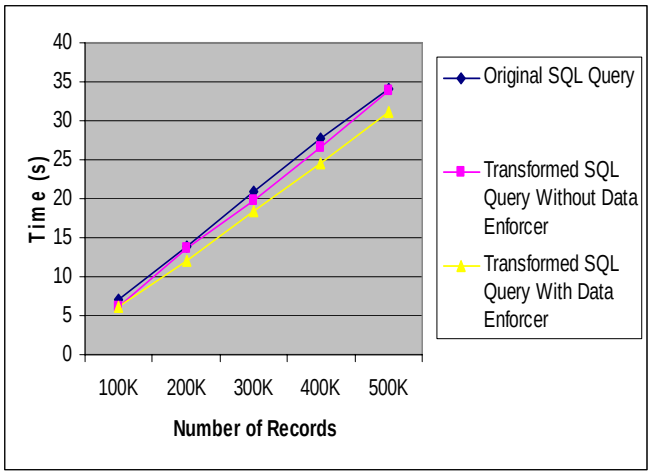

Fig. 9. Performance Tests: Results

To conclude, no noticeable loss of performance (considering the time spent between sending a query to a RDBMS and retrieving the last returned record) has been registered so far, on common SQL queries.

More tests and experiments are in progress on different varieties of SQL queries, involving also functions and stored procedures.

As next steps, we are also planning to: (1) explore the implementation of additional data transformation mechanisms, such as encryption and statistical transformation; (2) explore

the implications of post-processing query results to extend the current set of managed privacy constraints; (3) explore alternative way to intercept queries, without having to modify applications/services to use our driver. This might require operating at the network layer by running our Data Enforcer as a network service; (4) explore the enforcement of privacy policies on LDAP repositories, virtual and meta directories by building related data enforcers and author privacy policies.

VII. CONCLUSIONS

Privacy management is important for enterprises to ensure their compliance to laws and address customers' preferences and rights. Currently much is done via manual processes that makes them hard to comply.

This paper focuses on how to automate the enforcement of privacy policies on personal data, collected and stored by enterprises – by leveraging state-of-the-art enterprise identity management solutions.

We introduced a privacy-aware access control model to enforce privacy policies on personal data - including handling stated purposes for collecting data, checking data requestors' *Intent* against data purposes and enforcing data subjects' consent.

Based on this model, a working prototype has been implemented and integrated with a state-of-the art identity mangement solution, HP Select Access, integrated with its current (security-based) access control management capabilities. Our technology allows administrators to model personal data and graphically author related privacy policies together with traditional access control policies. Attempts to access personal data via queries are intercepted: queries are transformed (or blocked) to ensure that the returned result is consistent with stated privacy policies. Current performance results are encouraging.

Our technology is ready for commercial exploitation. Research and development work will continue to refine our technology and implement adiditional functionalities.

REFERENCES

[1] C. Laurant, "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/phr2003/ 2003

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

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

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

[5] G. Karjoth, M. Schunter, M. Waidner, "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

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

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

[8] IBM, "The Enterprise Privacy Authorization Language (EPAL), EPAL 1.1 specification", http://www.zurich.ibm.com/security/enterprise-privacy/epal/, IBM, 2004

[9] R. Agrawal, J. Kiernan, R. Srikant, Y. Xu, "Hippocratic Databases", http://www.almaden.ibm.com/cs/people/srikant/papers/vldb02.pdf, IBM Almaden Research Center, 2002

[10] IBM Tivoli Privacy Manager, "Privacy manager main web page", http://www-306.ibm.com/software/tivoli/products/privacy-mgr-e-bus/, 2005

[11] IBM Tivoli Privacy Manager, "online technical documentation", http://publib.boulder.ibm.com/tividd/td/PrivacyManagerfore-business1.1.html, 2005

[12] HP, "HP Select Federation - Product and Solution Overview", http://www.managementsoftware.hp.com/products/slctfed/, 2005

[13] ePok, "identity management solution - Trusted Data Exchange Server", http://www.epokinc.com/, 2005

[14] HP, "HP OpenView SelectAccess - Overview and Features", http://www.openview.hp.com/products/select, 2005

[15] J. Byun, E. Bertino, N. Li, "Purpose based access control for privacy protection in Database Systems", Technical Report 2004-52, Purdue University, 2004

[16] M. Casassa Mont, P. Bramhall, J. Pato, "On Adaptive Identity Management: The Next Generation of Identity Management Technologies", HPL-2003-149, 2003

[17] HP, "HP OpenView SelectAudit", http://www.managementsoftware.hp.com/products/slctaud/, 2006

[18] M. Casassa Mont, S. Pearson, P. Bramhall, "Towards Accountable Management of Privacy and Identity Information", ESORICS 2003, 2003

[19] M. Casassa Mont, R. Thyne, Pete Brmhall, "Privacy Enforcement with HP Select Access for Regulatory Compliance", HPL-2005-10, 2005

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

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

[22] PRIME, "Privacy and Identity Management for Europe, European RTD Integrated Project under the FP6/IST Programme", http://www.prime-project.eu.org/, 2004

[23] HP, "HP OpenView Select Identity – Overview and Features", http://www.openview.hp.com/products/slctid/index.html, 2005

[24] M. Casassa Mont, R. Thyne, P. Bramhall, "Privacy Enforcement for IT Governance: Doing it for Real ", TrustBus 2005, 2005

[25] Oracle, "Oracle Virtual Private Database", http://www.oracle.com/technology/deploy/security/db_security/htdocs/vpd.html, 2006