Towards Automated Privacy Compliance in the Information Life Cycle

Rema Ananthanarayanan, Ajay Gupta, and Mukesh Mohania

IBM India Research Lab, 4-C, ISID, Vasant Kunj,
New Delhi - 110 070, India

1 Introduction

Management of data is an increasingly challenging task for enterprises because of the increasingly diverse and complex requirements that come to bear from various fronts. One issue is the protection of the privacy of the information held, while ensuring compliance with various rules and regulations relating to data management. Privacy may be defined as the claim of individuals, groups and institutions to determine for themselves when, how and to what extent information about them is communicated to others. [1] In its broadest scope this is a problem that has challenged civil rights advocates the world over, at different times. However, with the growth of the web, the issue gains special significance in view of the following.

1. **Volumes of data involved:** Enterprises today hold large amounts of confidential information, in the form of transactional data, customer preferences gathered from web sites and other business data. Data warehouses on the petabyte scale are expected to be increasingly common, and large parts of this data would involve sensitive information about customers and employees. This could be in structured format, in relational tables, or in unstructured format, in the form of spreadsheets, emails and other reports.
2. **Intentional and unintentional exposure:** [1] cites many instances of accidental and intentional violation of privacy resulting from privacy accidents, ethically questionnable behaviour and lax security measures with respect to such data. The resulting litigation and adverse publicity have made companies increasingly sensitive to how such data is handled. Any exposure on this front directly affects the bottomline of companies and hence an increasing concern of any CIO is the effective protection of privacy data.
3. **Increasing rules and regulations:** A few high-profile cases of privacy abuse or neglect have led to increased awareness alround of the dangers of inadequate privacy protection. Consumers are increasingly demanding control over how the privacy information is used and safe-guarded. Regulations have come from both the indutry and governments. The former is an effort to build customer trust and goodwill, while the latter is to provide a measure of protection where self-regulation cannot be depended upon.

[1] This definition is attributed to Professor Alan Westin, Columbia University, 1967

T.S. Dillon et al. (Eds.): Advances in Web Semantics I, LNCS 4891, pp. 247–259, 2008.

Initial solutions to the problem of privacy data management were based at the level of each individual application, and mostly introduced manual processes or at best semi-automated processes. However, these solutions are inefficient for handling the high volumes of data that are increasingly becoming common. Further, they do not scale across different applications in the enterprise. Hence it becomes very urgent to drive the search for automated solutions, which would further be application-neutral. These automated solutions need to protect the data at every stage independent of which application accesses the data.

In this work, we look at some of the issues associated with the problem of privacy protection, and present some directions of work that would help automate the solution to the problem. Our work is organised as follows. Section 2 discusses the background to the problem in terms of how customer requirements have resulted in regulations necessitating right business processes for efficient handling. In section 3 we discuss some of the IT-related standards that have evolved in response to the demands of privacy protection of information from enterprises, and some client-side solutions to the problem. In section 4 we discuss some server-side solutions to the problem. Some of these architectures are currently in use in industrial products, while others are proposed for new privacy rule structures that may become more common. We conclude in section 5 highlighting some current open issues and future directions of work.

2 Background

Personal identifiable information (PII) may be defined as any information such as an identification number, name and phone number or address that helps to identify the individual to whom the information pertains. Online transactions have increased the amount of personal identifiable information that is available to enterprises, from their customers. This repository of information is very useful for mining applications that mine for patterns and trends, to study individual preferences and customise offerings to individuals based on their purchasing pattern or preferences and related applications. However, increasingly customers are unwilling to share personal information unless they are assured that some safeguards are in place for protecting this information and controlling its use. One of the earliest and well-known studies to assess user attitudes about online privacy is [2]. One of the main findings of this report was that users registered a high level of concern about privacy in general and on the Internet. The study findings reported that 17% of the web users were privacy fundamentalists who were extremely concerned about any use of their data and generally unwilling to provide data even when protection measures were in place. 56% were observed as a pragmatic majority, who had specific concerns which were generally reduced by privacy protection measures such as privacy policies on Web sites. The balance 27% were observed as being marginally concerned and generally willing to provide data to Web sites under almost any condition, although they expressed a mild general concern about privacy. In a different study specifically focusing on health information, [3] some key findings were that customers rate personal

healthcare and financial information the most sensitive types of consumer personal information. While 80% of the customers visit health sites for information, they express high concerns about privacy and security in their surfing. Further, because of their privacy concerns, many consumers visiting these sites do not share their personal data and hence fail to take full advantage of these sites.

In view of the above, enterprises started publishing privacy policies that stated how the information gathered on their site would be used or would not be used.This served as a self-regulatory exercise to convince users that the enterprise cared about protecting user privacy. Most of these privacy statements drew broadly on the fair information practices (first articulated in the US Department of Health, Education and Welfare report, also known as the HEW report [4]), whose highlights are descibed in table 1.

Table 1. Fair information practice principles

Notice/Awareness	Web sites should provide full disclosure of what personal information is collected and how it is used.
Choice/Consent	Consumers at a web site should be given choice about how their personal information is used.
Access/Participation	Once consumers have disclosed personal information, they should have access to it.
Integrity/Security	Personal information disclosed to web sites should be secured to ensure the information stays private.
Enforcement/Redress	Consumers should have a way to resolve problems that may arise regarding sites' use and disclosure of their personal information.

Apart from these, privacy policies in general have also drawn from the OECD guidelines. These are guidelines on international policy for the protection of privacy and transborder flows of personal information [5] drawn up by the Organization for Economic Co-operation and Development (OECD). These guidelines represent a consensus on the general guidance concerning the collection and management of personal information. The main driving principles laid down by the OECD for data collection are collection limitation, data quality, purpose specification, use limitation, security safeguards, openness, individual participation and accountability. In the next section we discuss some specific policy frameworks that draw from these guidelines.

3 Policy Frameworks

3.1 Frameworks

The Platform for Privacy Preferences (P3P) is a standard established by the WorldWide Web Consortium (see http://www.w3.org/P3P) that served as a first step in the front-end for building customer trust and goodwill. P3P enables web site owners to define and publish the privacy policy followed by the web site, in manual and machine-readable formats. This allows user-agents to read

and interpret privacy policies automatically, on behalf of users. Users may then decide what information to share and what not to disclose, based on the site's privacy policy. Though P3P does not state how the policies will be enforced, leaving it to the individual enterprises, P3P has been widely adopted as a front-end measure by companies hoping to establish trust and build confidence in the users. These standards have served as a basis for client-side implementations of privacy policies. For instance, browsers such as Netscape 7.0 and Internet Explorer 6.0 support the standard by giving users more control over cookies and allowing the user to easily view the web site's privacy policy.

However, publishing good privacy polices does not automatically translate to processes that enforce these policies. One practice that has developed has been for individual web sites to obtain privacy seals from third-party seal providers, who are established and reputed companies in this line. Privacy seal companies such as TRUSTe (http://www.truste.org) and BBBOnline (http://www.bbbonline.com) provide privacy seals which are images displayed on the Web sites of companies that register for these seals. These seals basically ascertain that the companies care about user privacy and hence have registered with the seal providers who now have the responsibility to

1. Review their privacy policies periodically and ensure that it is in line with the latest applicable privacy legislation;
2. Ensure that the Web site accurately discloses its data collection activity;
3. Audit the privacy-related business processes periodically to ensure that the stated policies are being followed.

Another standard which defines how privacy policies may be written, in order to be easily enforceable is XACML. XACML or extensible access control language [6] is an XML-based language for access control, standardized by the Organization for the Advancement of Structured Information Standards (OASIS). XACML describes both the access control policy language (to represent access control policies of who can do what and when) and a request/response language to express queries about whether a particular access should be allowed (request) and describes answers to those queries (responses).

3.2 Privacy Compliance at the Back-End

At the back end, works such as the Hippocratic data bases [1] draw upon and extend the OECD guidelines, to define a set of standards in terms of data access and protection, that would comprise the Hippocratic database. Here it is envisaged that in addition to the existing capabilities of databases today, such as the ability to manage persistent data and the ability to access a large amount of data efficiently, data repositories in future would also need to ensure that they adhere to a certain set of standards in terms of data access and protection. These privacy principles, also borrowing from the OECD guidelines, include broadly purpose specification, consent, limited collection, limited use, limited disclosure, limited retention, accuracy, safety, openness and compliance. In the next section we discuss the impact of some of the above on the implementation

of privacy-compliant solutions for data management. Traditionally Role-Based Access Control has been widely used for controlling access to information [7]. Users are assigned roles based on their job functions, and access or deny rights to data is based on the user roles. [8] discusses language constructus and implementation designs for fine grained access control that can be applied to current database management systems, to transform them to their privacy equivalents. These constructs include column restrictions, row restrictions and cell restrictions; the work also describes how privacy rules in P3P can be translated into proposed constructs. The usage control or UCON model [9] encompasses traditional access control along with trust management and digital management and presents a future direction for access control over the traditional model. UCON enables fine-grained control over digital objects, for instance, print once as opposed to unlimited prints. Subjects, objects and rights are the core components while other components are involved in the authorization process. Obligations are defined as mandatory requirements that a subject has to perform after obtaining or exercising rights on an object. An example is, a consumer may have to accept metered payment agreements before obtaining the rights for the usage of certain digital information. By including the notion of obligation with authorization, the model provides for better enforcement on exercising usage rights for both provider and consumer subjects. UCON does not state how the obligations are enforced.

While these schemes apply to data residing in individual applications, subsequent directions have been to design solution architectures that apply at an enterprise level, to ensure compliance with the various privacy rules, for data accessed by different applications. One approach is to define middleware that abstracts privacy and data-handling rules from applications and applies it across the various systems centrally. In the subsequent sections, we discuss these schemes.

4 Middleware for Privacy Protection

Standards such as P3P provide for privacy compliance at the client side, ensuring adherence in terms of privacy seals and audits. Privacy enforcement at the database level enforces privacy restrictions on the server side. We now discuss some schemes for enforcing privacy through middleware designed for this purpose. The middleware abstracts privacy and data-handling rules from applications and applies it across the various systems centrally.

4.1 Information Life Cycle

Privacy policies mandate what information is collected, how it is used and how long it is retained. As such, the policy could impact the data at different stages of the information life cycle. Hence we describe the approach in terms of the various stages in the life cycle of the data item. Table 2 gives examples of different legal directives that impact data retention at different points the life cycle of the data.

Table 2. Instances of regulations affecting data management at different stages in the lifecyle of the data item

Requirement	Instance of regulation
Mandated disclosure	Sections of the Sarbanes Oxley Act mandate disclosure of different classes of information within different time periods which may immediately, the next business day, or on a quarterly basis.
Mandated nondisclosure	The Gramm-Leach-Bliley Financial Privacy Act limits instances in which financial institutions may disclose non-public personal information about a customer to non-affiliated third parties.
Mandated retention	As stipulated in the Health Insurance Portability and Accountability Act 1996, (HIPAA) hospitals and healthcare providers must maintain medical records as well as billing records on medicare, medicaid and maternal and child health for at least 6 years.
Disposal	For operational efficiency, most enterprises define disposal schedules for disposable information such as mass communications, draft versions of documents and duplicates of documents. For instance, the rule may state that disposable information can be disposed within 90 days or sooner, unless it is specifically required for business purposes, when it may be retained for longer periods, but not exceeding 2 years.
Mandated non-collection	The Children's Online Privacy Act mandates that web sites collecting personal information from children under the age of 13 may do so only if they have obtained verifiable parental consent before the collection, use or disclosure of any personal information. Else such information may not be collected.

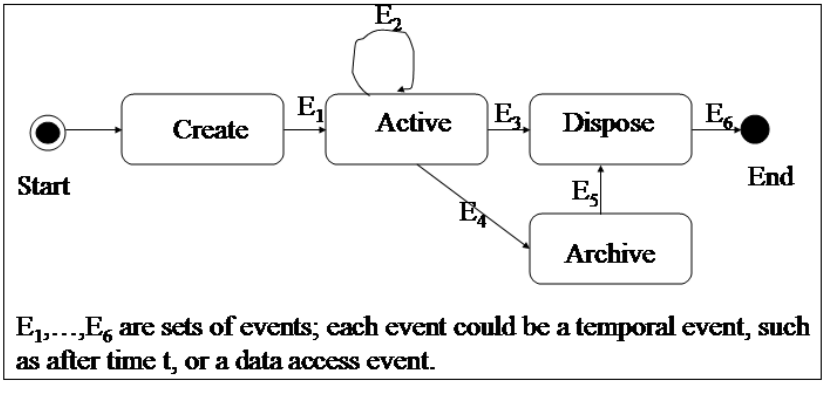

Fig. 1. State diagram for a data item in an enterprise

Figure 1 shows a simplified view of the life cycle of a data item or information resource, in the enterprise. This figure defines four states,in addition to a 'Start' state and an 'End' state. Information may be added to different repositories when people register and submit information, when sales transactions happen and at other points in time. The creation of such information may itself be within the scope of the privacy policy. Transitions from one state to another may be effected based on user tasks such as reading or updating documents and temporal events such as after a certain time period, or after m accesses by users. The above figure is only one possible state diagram. The actual state diagram would be defined by the record-keeping and data-handling rules of the enterprise. Further, some of the states in the figure, such as the 'Active' state, could themselves be viewed as comprising additional states.

4.2 Artefacts of a Privacy Policy

In this section we define in detail the various artefacts of a typical privacy policy. We have been guided by a policy definition similar to the XACML standard. Each privacy rule defines whether a data access is allowed or denied, based on the kind of data being accessed, the user role, the intent and intended action. In addition, each rule may be permitted only if certain conditions in the environment are met, and further, each rule may also mandate one or more obligations on the system. Figure 2 shows the artefacts of a typical policy and the various components are defined below.

Definition 1. *A privacy policy comprises a set of privacy rules.*

A rule in a privacy policy specifies whether a specific request is to be allowed or denied, in the context of some conditions in the environment evaluating to true; the rule may optionally also mandate one or more obligations. Formally, we define a rule as under.

Definition 2. *A rule is four tuple, comprising an event, a condition, a permission and optionally, one or more obligations.*

Each of these components is now defined below. An event relates to those components of a user request that are used to determine whether the request is allowed or denied.

Definition 3. *An event is a 4-tuple, comprising the user category, the data category, the action and the purpose associated with a user request.*

An event E is represented as $E = <u,d,a, p>$ where u is the user category to which the user (who is requesting the data) belongs, d is a data category to which the data item belongs, a is the action that the user wishes to perform on the resource, and p is the purpose for which the resource is being accessed. The following are some of the other terms that we will use in the rest of the paper.

User category: Each user of the data in the system is assigned to one or more user categories, based on the user roles. The user categories could be hierarchical.

Data category: Each data item that is to be privacy-protected is assigned to a specific data category. The data categories could be hierarchical.

Action: Users may access the data items to perform one or more of the specified set of actions. A representative set of actions could be {`'View', 'Modify', 'Delete'`}.

Purpose: Users may access the data item for a specific purpose, from one of the set of purposes specified. A representative set of purposes could be {`'Marketing', 'Research', 'LegalPurposes', 'Administrative'`}.

Data item: A data item is an entity that is to be privacy protected, ie., whose access is controlled by one or more rules in the privacy policy. Each data item belongs to some data category. Examples of data items include emails, facsimile documents, customer records and scanned images (of xrays, for instance, when the domain is the health industry).

The privacy rules come into effect when a user makes a request on the system.

Definition 4. *A user request is a 4-tuple comprising the user id, the data item being requested, the intended action and purpose.*

Note that each user request maps internally to an event, where the user is mapped to the user category and the data item is mapped the data category. Some more terms that we use are:

Condition: This is a clause associated with some rules, which needs to be true, for the rule to 'Allow' access. Examples of conditions include *If the user is above 18 years of age*, *If the user has consented*, ...

Permission: This is a binary predicate that states whether the user request is allowed or denied. It can take one of the two values, 'Allow' and 'Deny'.

Obligation: An obligation is a task optionally associated with some rules, that specifies one or more actions that need to be mandatorily performed after the user request is fulfilled. Some examples are *Log all accesses to this data* or *Email the customer if this information is used for marketing.*

Figure 2 is an object model representation of the key objects of interest, and their relationships, in the context of a privacy policy as we have defined above. Note that in theory a large number of events are possible, but in practice the system is interested only in events which are to be specifically permitted or specifically denied. All other events are not usually catalogued in the policy, but assumed to take on a default permission of 'Allow' or 'Deny'. Hence in practice the number of events that figure in the policy are much smaller than the total number of events based on the possible combinations of user groups, data items and other entities that define an event. Further, as we can see, a user request is only a specific instance of an event.

Figure 3 shows a view of the data management in terms of the privacy policy enforcement. As we can see from the figure, the relevant rule in the privacy policy is associated with the data item at the time of the data creation, which could be at the time of the data collection, for instance. If any rule applies to this data item, all subsequent accesses to this data item are monitored. A rule is

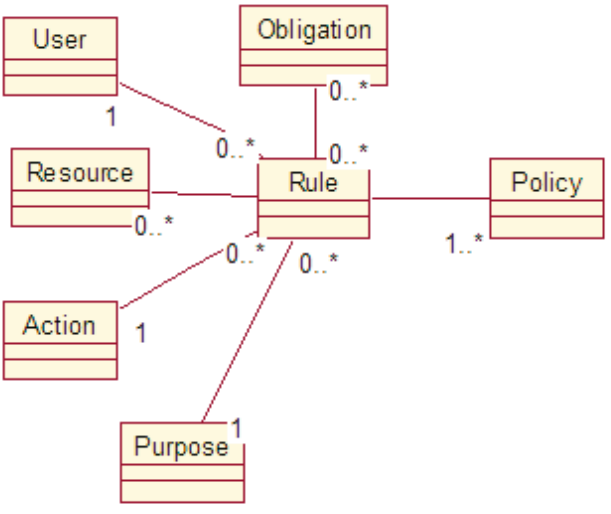

Fig. 2. Object model of the rules in a policy

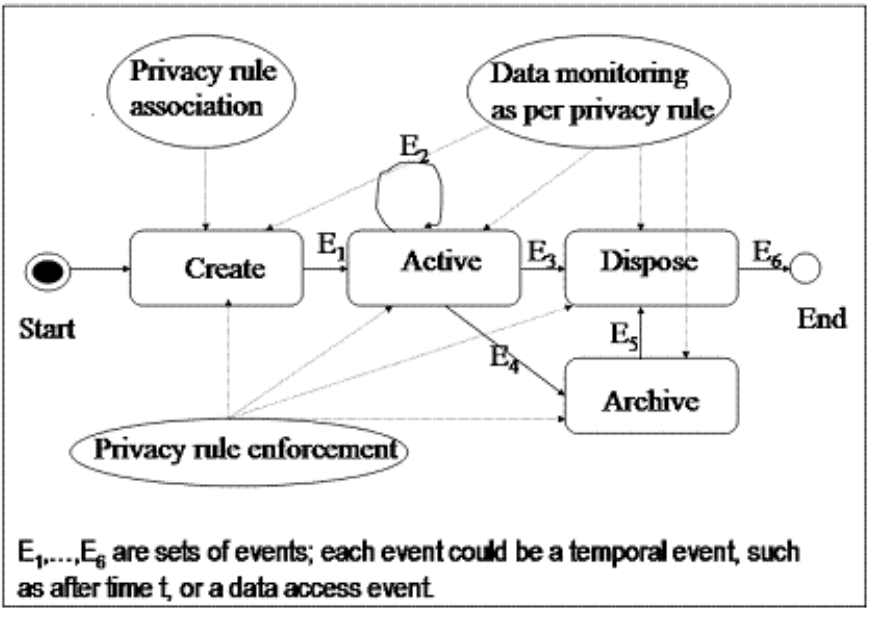

Fig. 3. States and applicable policy management

fired on the occurrence of an event. An event could be temporal, such as at time t or after time t_1. An event could be an access event such as on the first access of a data item, on every access of a data item, or on the n'th access of a data item. On occurrence of an event the relevant execution rule is fired, and the data

item stays in its current state or is transitioned to the next state, depending on the privacy rule that is being applied. We define the middleware necessary to enforce rule compliance in such a framework, in the next section.

4.3 Components of the Middleware

In this section we describe the various components that would be required to perform data management at the various stages of the information life cycle, as shown in figure 3. These components and their functionality are:

1. A policy translator that translates the high-level language policy to a machine-readable format, for example, XML. (The functionality of this component is outside the scope of the current work and hence we do not discuss this further.)
2. A policy association module that associates each data item with the applicable rule/s in the policy.
3. A request interceptor, that monitors the data items for the occurrence of events. An event could be temporal or based on data access.
4. An event handler that is invoked by the request monitor whenever an event of interest occurs. The event handler basically reports whether the specific access request is allowed or denied, in the case of access events. If applicable, the event handler updates the state of the data item as relevant.

Figure 4 presents a possible architecture for privacy compliance end-to-end with the above components, extending the notions in figure 3. The functionality of the various components may be defined as follows.

1. *Rule association:* The data items in the various repositories to be protected may be grouped into different categories for which the privacy rules are defined. The rule association step associates each data item with zero, one or more rules, based on the specific category to which the rule belongs. This may be done at setup time, and subsequently may need to be done whenever new rules or new data items are added to the system. If the rules are specified in a structured or semi-structured format, then it should be possible to do this step in a semi-automated manner.
2. *Request interceptor:* The request interceptor intercepts any user requests to the data in the various applications. This may be achieved by building a common interface that is accessed by the request interceptor in the front-end. At the back-end, this interface invokes each application-specific API, for each specific application. The only requirement is that each of the repositories need to expose APIs for access control and subsequent querying, which can then be invoked by the request interceptor. Any user request is made to the request interceptor. The request interceptor than passes on the necessary information to the event handler (described subsequently) and based on the reply from the event handler, determines whether the access is to be allowed or not. If yes, then the request is made to the API of the specific application and the results returned to the user.

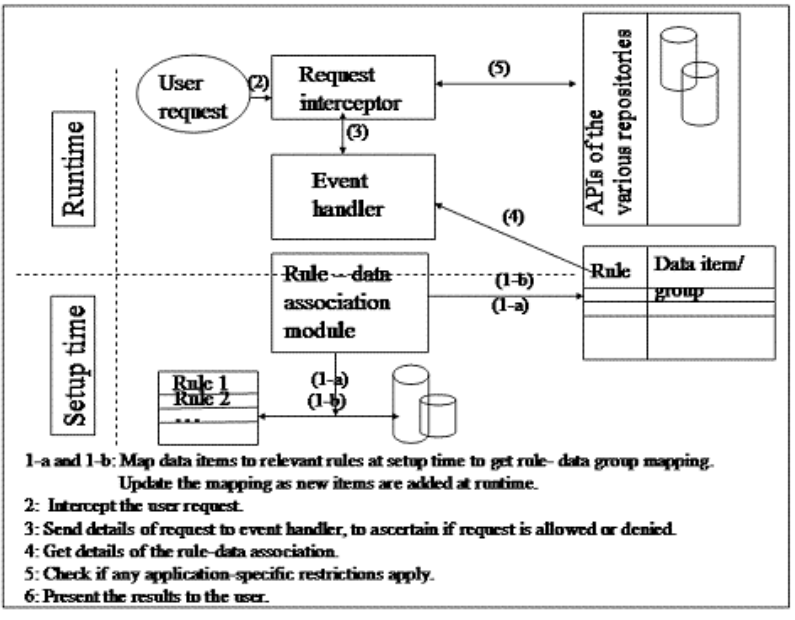

Fig. 4. Architecture of a simple privacy rule enforcement system

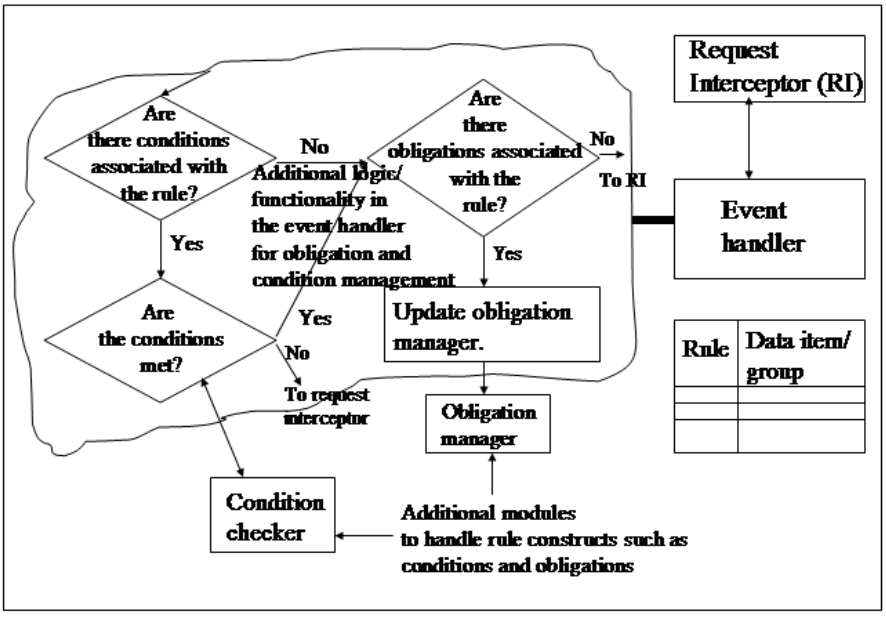

Fig. 5. Architecture of a rule enforcement system with associated conditions and obligations

3. *Event handler:* The event handler determines at runtime whether any rule is applicable for a data access. The request interceptor intercepts the user request and sends the details to the event handler. The event handler determines if one or more rules are applicable for the access event defined by this data request. The event handler performs this step based on the information available after rule association in the first step above. Based on this, the event handler returns an Allow or Deny ruling to the request interceptor. The system may have a top-level policy rule that states how access requests for events that are not specifically allowed or denied are to be handled.

In this architecture, we have considered simple privacy rules that are associated with an Allow or a Deny ruling, and are not associated with any conditions or obligations. In this case, it is possible to combine the functionality of the request interceptor and the event handler. However, if we consider privacy rules that are associated with conditions and obligations, then we can achieve enforcement of the same by including additional modifications in the event handler. This is shown in figure 5. The functionality of the event handler is extended to check for the validity of the condition. An additional component, the obligation manager, is also required to handle obligations. Here, when an event occurs, the event handler has to perform the following additional tasks.

1. If there are any conditions associated with the rule, then it needs to be checked whether the condition is met, and the data access allowed only if the condition is met.
2. If there are any obligations associated with the rule, then the obligation manager needs to be informed of the data access event and other needed information to enforce the obligation execution.

These additional details are shown in figure 5.

In this section we have discussed some possible architectures for privacy protection of data, using a middleware approach. The advantage of such as approach is that it enables the architecture to be uniformly applied across multiple content repositories, irrespective of their underlying format. The repositories only need to expose APIs for requesting data, and for the repository-specific access control that may be in place over and above the privacy policy. A further advantage is that no changes need to be made to the way the data is currently stored.

5 Conclusions

Protection of privacy information is a very important challenge across enterprises. Most solutions that exist today are application-specific or content-specific and hence do not scale readily or lend themselves to automation readily. However automated solutions are a very important requirement in view of the increasing volumes of data that enterprises handle, and the increasing exposure that enterprises face on the risk and compliance front. In view of these, we have presented a possible architecture for privacy protection at the back-end. Our initial

architecture is for a system comprising simple privacy rules that specify which user groups can access which data items or data categories. The advantage of this architecture is that it can scale readily across different applications in the enterprise and is independent of the type of content or repository. Further it lends itself readily to automation with some minor additions to the architecture shown. Subsequently, we have presented extensions to the same architecture for privacy rules that may be more complex, for instance, including conditions and obligations. These would be essential constructs of privacy rules in future, in view of the increasing emphasis on risk and compliance management. Further, our architecture also aims to integrate the privacy management along with the information life-cycle management, since our contention is that future data management systems would be closely integrated with the information life cycle.

A number of open issues remain in related areas. At the top-level, the rules are framed in a business context, in a high-level language. However, for full automation, we need a way of translating the rules to a machine-understandable format. Currently this is a manual step in the process, which would need to be done for introduction of any new rule in the system. Further, conditions and obligations are increasingly a part of the policy framework. Hence the automated system needs to handle automated checking of conditions and automated obligation execution, for full scaleability.

References

1. Agrawal, R., Kieman, J., Xu, R.Y.: Hippocratic databases. In: Proceedings of the 28th VLDB Conference (2002)
2. Cranor, L.F., Ackerman, M.S.: Beyond concern: Understanding the net user's attitude about online privacy. Technical report TR 99.4.3, AT&T Labs Research (1999)
3. Westin, A.F.: How the public views health privacy: Survey findings from (1978–2005), `http://www.pandab.org/HealthSrvyRpt.pdf`
4. Records, computers and the rights of citizens: report of the secretary's advisory committee on automated personal data systems (1973), http://www
5. Oecd guidelines on the protection of privacy and transborder flows of personal data (1980), `http://www.oecd.org`
6. Xacml 2.0 specification set, `http://www.oasis-open.org/committees/xacml/`
7. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models, vol. 29(2), pp. 38–47 (1996)
8. Agrawal, R., Bird, P., Grandison, T., Kiernan, J., Logan, S., Rjaibi, W.: Extending relational database systems to automatically enforce privacy policies. In: 21st International Conference on Data Engineering (ICDE 2005), pp. 1013–1022 (2005)
9. Park, J., Sandhu, R.: Towards usage control models: Beyond traditional access control. In: Proceedings of the 7th Symposium on Access Control Models and Technologies (2002)