A Semantic Privacy-Preserving Model for Data Sharing and Integration*

Yuh-Jong Hu
ENT Lab., Dept. of CS
National Chengchi University
Taipei, Taiwan, 11605
hu@cs.nccu.edu.tw

Jiun-Jan Yang
ENT Lab., Dept. of CS
National Chengchi University
Taipei, Taiwan, 11605
98753036@nccu.edu.tw

ABSTRACT
In this paper, we encompass and extend previous ontology-based data integration system. A semantic privacy-preserving model provides authorized view-based query answering over a widespread multiple servers for data sharing and integration. The combined semantics-enabled privacy protection policies are used to empower the data integration and access control services at the virtual platform ($\mathcal{VP}$). The ontology mapping and merging algorithm with a local-as-view (LAV) source description that creates a global ontology schema at the $\mathcal{VP}$ by integrating multiple local ontology schemas for data sharing. The perfect rules integration of datalog rules enforces the data query and protection services. Semantics-enable policies are combined together at the $\mathcal{VP}$, but the access control criteria specified in each server are still satisfied. Therefore the soundness and completeness of data sharing and protection criteria are ensured to support the validity of policy combination. This guarantees the trustworthiness of data sharing and protection services in multiple servers.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*query formulation*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*data sharing*; K.4.1 [**Computers and Society**]: Public Policy Issues—*privacy, regulation*

General Terms
WWW, Semantic Web, Database

Keywords
data sharing and integration, semantics-enabled policy, privacy protection, query rewriting, ontology and rule

1. INTRODUCTION
Large enterprises spend a great deal of time and money on data (or information) integration [3]. Data integration is the problem of combining the data from autonomous and heterogeneous sources, and providing users with a unified view of these data through so called global (or mediated) schema. The global schema, which is a reconciled view of the information, that provides query services to end users. The design of a data integration system is a very complex task, which includes several different issues: heterogeneity of the data sources, relation between the global schema and the data sources, limitations on the mechanisms for accessing the sources, and how to process queries expressed on the global schema, etc [11].

Three approaches have been proposed to model a set of *source descriptions* that specify the semantic mapping between the source schema and the global schema. The first one, called global-as-view (GAV), requires that the each concept in the global schema is expressed in terms of query over the data sources. The GAV deals with the case when the stable data source contains details not present in the global schema so it is not used for dynamically adding or deleting data sources.

The second one, called local-as-view (LAV), requires the global schema to be specified independently from the sources, and the source descriptions between the stable global schema, such as ontology and the dynamic data sources are established by defining each concept in the data sources as a view over the global schema [10] [26]. LAV descriptions handle the case in which the global schema contains details that are not present in every data sources.

The third one, called global-local-as-view (GLAV), a source description that combines the expressive power of both GAV and LAV, allowing flexible schema definitions independent of the particular details of the data sources [14] [30]. The data integration system uses these different source descriptions to reformulate a user query into a query over the source schemas. However, data sharing and integration are hampered by legitimate and widespread privacy concerns so it is critical to develop techniques that enables the integration and sharing of data without losing a user's privacy [12].

Privacy protection policies represent a long-term promise made by an enterprise to its users and are determined by business practice and legal concerns. It is undesirable to change an enterprise's promises to customers every time an internal access control rule changes. If possible, we should enable the integration of Platform for Privacy Preferences (P3P) and Enterprise Privacy Authorization Language (EPAL) policies to provide accountable and transparent information processing for data owners to revise their data usage permissions [2].

Although many organizations post online privacy policies, they must realize that simply posting a privacy policy on their websites does not guarantee true compliance with existing legislation. Following the OECD's Fair Information Principles (FIPs)[1], an organization should provide norms of personal information process for its data collection, retention, use, disclosure, and destruction. An organization must also be accountable for its information possession and should declare the purposes of information usage before collection. Moreover, an organization should collect personal information with an individual's consent and disclose personal information only for previously identified purposes.

In this paper we are addressing the following research issues. More detailed modelling and implementation will be shown in the later sections.

- Data sharing and protection services are considered in a large number of servers. The incentives for using the virtual platform ($\mathcal{VP}$) is to avoid solving the complex pair-wise problem of ontology matching and rule integration between these servers. Therefore a unified global data sharing and protection service can be achieved at the $\mathcal{VP}$.
- Privacy protection policies are expressed as a combination ontology and rule, i.e. $\mathcal{O}+\mathcal{R}$, where ontology $\mathcal{O}$ includes `TBox` schema and `ABox` instances, and rules $\mathcal{R}$ include deductive rule set ($\mathcal{RS}$) and facts ($\mathcal{F}$). Data sharing and protection in multiple servers are achieved through a combination of semantics-enabled formal protection policy ($\mathcal{FPP}$).
- The challenge of designing a semantic privacy protection model is to ensure a *soundness* and a *completeness* of data sharing and protection in multiple servers. For the *soundness* criterion, we do not allow unintended data being released to the data users through the global policy schema ($\mathcal{GPS}$) at the $\mathcal{VP}$. Otherwise, it violates the privacy protection policies. As for the *completeness* criterion, we do not miss any eligible shared data when a user asks for a data request service at the $\mathcal{VP}$. Therefore, shareable data obtained at the $\mathcal{VP}$ should equal data obtained directly from each server.

Each enterprise server declares its P3P privacy protection policies that takes into account the FIPs criteria (see Figure 1). Then EPAL policies are established in each site, corresponding to the P3P [24]. For each data request, the data handling and usage controls are based on the EPAL policies. However P3P and EPAL lack formal and unambiguous semantics to specify privacy protection policies so they are limited in the policy enforcement and auditing support for the software agents. One of the research challenges for the online privacy protection problem is to develop a privacy management framework and a formal semantics language to empower agents to enforce privacy protection policies. Agents must avoid any policy violation of each data request. We attempt to establish a semantic privacy protection model to address this issue. Each server shares its collected data with other servers but without breaking the original data usage commitment to its clients [25].

The contributions of this paper are twofold. We first offer a three layers semantic privacy-preserving model which encompasses and extends the existing work on data sharing and integration by using a combination of ontology and rule for the representation of privacy protection policies. In particular, we define a formal policy using ontology for privacy protection concept descriptions and rule for data query and access control services. Then we focus on solving the soundness and completeness of query rewriting problem using a perfect ontology merging and a perfect rule integration from the local formal protection policies. Followed by each possible data query at the $\mathcal{VP}$, we briefly demonstrate how the soundness and completeness criteria for privacy protection data integration can be achieved using this semantics-enabled privacy-preserving model.

The paper is organized as follows. In section 2, we present a semantic privacy-preserving model as a framework for data sharing and integration services. In section 3, we define a formal policy combination as an integration of formal policies from autonomous data sources. Each formal policy is composed of ontologies and rules for each independent data source. A privacy protection policy is a type of formal policy used for specifying a data usage constraint from a data owner. In section 4, we formally define a formal policy combination in terms of ontology mapping, merging, and alignment. Then we demonstrate how a perfect rule integration is used for query rewriting at the $\mathcal{VP}$ corresponding to each local schema. In section 6, we briefly prove the soundness and completeness of privacy-preserving data sharing and integration based on this semantic privacy-preserving model. We conclude with related work and discussion in the last two sections.

2. A PRIVACY-PRESERVING MODEL

A semantic privacy protection model is proposed with three layers, where the bottom layer provides data sources from the relational databases, the middle layer provides a semantics-enabled local schema for each independent service domain. The top layer is served at the $\mathcal{VP}$, which provides a unified global view of privacy-preserving data sharing and integration services (see Figure 2).

We have a merged global ontology schema created by mapping and aligning local ontology schemas with a LAV source description from multiple local schemas in the middle layer. The idea of using description logic (DL) to model the local and global schemas is to empower the ontology's abstract concept representation and reasoning capabilities. A query

[1]See http://www.privacyrights.org/ar/fairinfo.htm

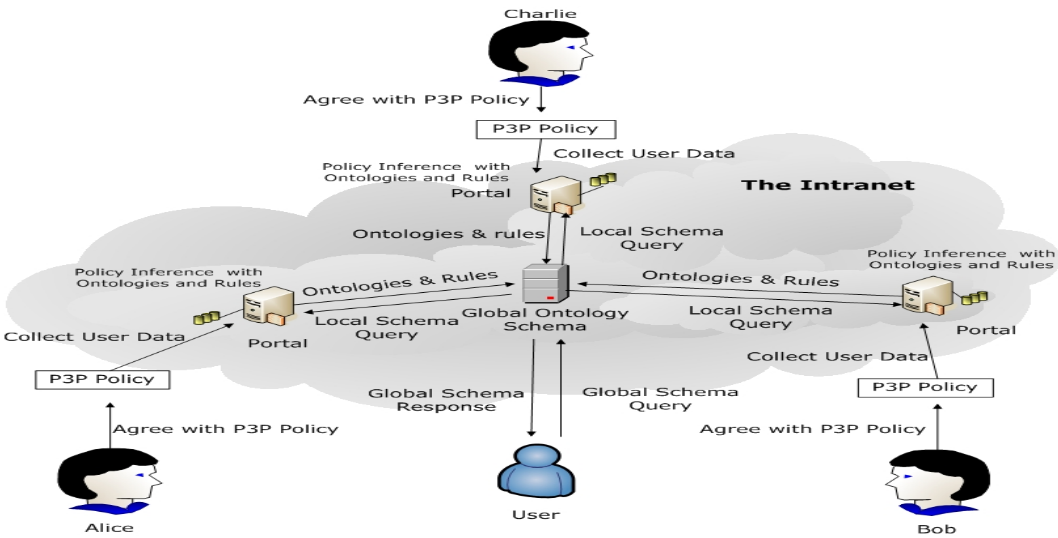

Figure 1: A semantic privacy protection model extended from the integration of P3P and EPAL for data sharing and protection in multiple servers

is defined as an SQWRL datalog rule in the SWRL-based policy to access to a global ontology [31]. Each SQWRL data service query for a global ontology at the $\mathcal{VP}$ is mapped to multiple queries as SQWRL datalog rules for each local schema. This is a LAV query rewriting service which has been investigated in databases but it is largely unexplored in the context of DL-based ontologies [14].

2.1 Formal Privacy Protection Policy

A policy's explicit representation in terms of ontologies or rules depends on what the underlying logic foundation of your policy language is. If your policies are created from DL-based policy language, such as Rein or KAoS, then ordinary policies are shown as TBox schema and ABox instances. Otherwise, policies created from LP-based policy language, such as EPAL or Protune ordinary policies are a set of rules with predicates of unary, binary, or ternary variables and facts [5].

In the SemPIF framework [21], we define Policy Interchange Format (PIF) to follows W3C $\mathcal{O}+\mathcal{R}$ standards [6] and strives to provide a mechanism for agents to preserve different policy syntax and semantics throughout its policy integration and interchange. In addition, agents can use meta-PIF, providing further management and reconciliation services of PIF-enabled multiple policies across various domains. In this paper, we apply the SemPIF framework for the privacy-preserving data integration through a combination of formal policies.

A formal policy ($\mathcal{FP}$) is a declarative expression corresponding to a human legal norm that can be executed in a computer system without causing any semantic ambiguity. An $\mathcal{FP}$ is created from a policy language ($\mathcal{PL}$), and this $\mathcal{PL}$ is shown as a combination of ontology language and rule language . Therefore, an $\mathcal{FP}$ is composed of ontologies $\mathcal{O}$ and rules $\mathcal{R}$, where ontologies are created from an ontology language and rules are created from a rule language.

A formal protection policy ($\mathcal{FPP}$) is an $\mathcal{FP}$ that aims at representing and enforcing resource protection principles, where the structure of resources is modelled as ontologies $\mathcal{O}$ but the resources protection is shown as rules $\mathcal{R}$.

A privacy protection policy shown as an $\mathcal{FPP}$ is a combination of ontologies and rules, e.g., $\mathcal{O}+\mathcal{R}$, where DL-based ontologies, such as OWL-DL ontologies provide a well-defined structure data model for data sharing, while Logic Program (LP)-based rules, such as datalog rules provide further expressive power for data query and protection. There are numerous $\mathcal{O}+\mathcal{R}$ combinations available for designing privacy protection policies, such as SWRL [20], and OWL2 RL [17]. Each $\mathcal{O}+\mathcal{R}$ combination implies what expressive power we can extract from ontologies for the rules and vice versa.

The SWRL is one of the $\mathcal{O}+\mathcal{R}$ semantic web languages suitable for a policy representation in the privacy protection model. But this is not an exclusive selection. Other $\mathcal{O}+\mathcal{R}$ combinations, such as CARIN, OWL2 RL are also possible for modeling formal privacy protection policy whenever their underlying theoretical foundations and development tools are available. We fully utilize the SWRLTab development tools and SQWRL OWL-DL query language [31] in the Protégé to model and enforce semantic privacy protection policies.

We face a research challenge of combining SWRL-based privacy protection policies from multiple servers to ensure the soundness and completeness of data sharing and protection criteria. Another challenge is to solve the policy's syntax and semantics incompatibility when we allow policy combination in multiple servers. SWRL is based on the classical first order logic (FOL) semantics that mitigates a possible semantic and syntax inconsistency when policies come from different servers.

But we still face a background policy inconsistency problem when default policy assumptions vary between different

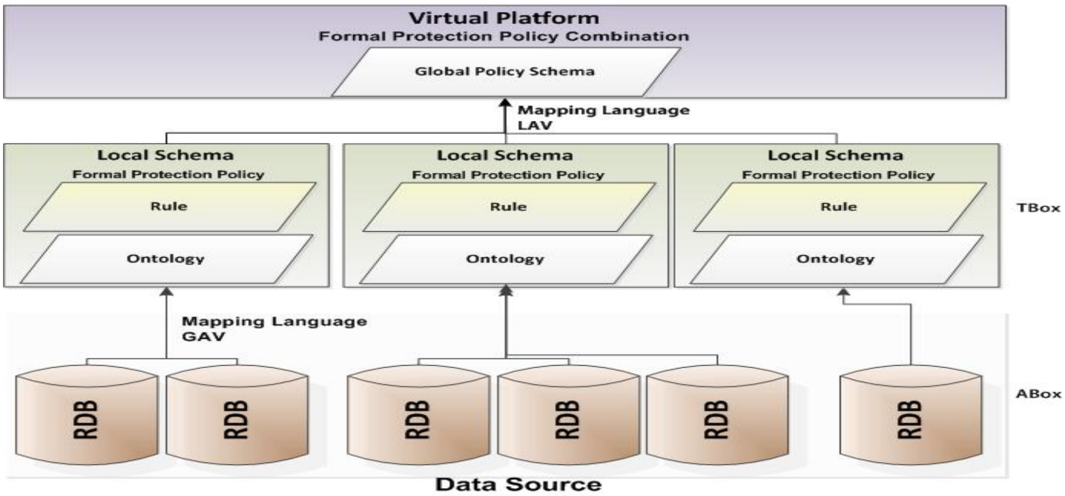

Figure 2: A semantic privacy protection model

servers. For example, one server uses open policy assumption, where no explicit option-out for data usage means option-in, but the other server uses closed policy assumption, where no explicit option-in for data usage means option-out. We avoid this kind of policy inconsistency by requesting all sites to use a uniform policy assumption, and to collect option-in data usage choices from users whenever multiple policies are integrated.

Previous studies for policy combination did not consider solving the problem of merging multiple schemas and integrating access control rules from multiple servers [4] [28]. In this paper we propose a semantic privacy protection model that allows flexibly combining `TBoxes` of privacy protection policies without moving `ABox` instances from its original data source until a data request service is initiated (see Figure 3). Therefor the global ontology `TBox` schema and rules created at the $\mathcal{VP}$ have the latest updated incoming data from each server when a user asks a query.

Data integration aims at providing unified and transparent access to a set of autonomous and heterogeneous data sources. The semantic privacy protection model providing global ontology schema for data sharing is similar to the data integration problem solved by $DL - Lite_A$ ontologies shown in [8]. Here we are also focusing on data protection besides data sharing and integration.

The goal of ontology-based data integration in $DL - Lite_A$ is to provide a uniform access mechanism to a set of heterogeneous relational database sources, freeing the user from having the knowledge about where the data are, what they are stored, and how they can be accessed. The idea is based on decoupling information access from its relational data storage so users only access the conceptual layer shown as ontology, while the relational data layer, hidden to users, manages the data.

Compared with $DL{-}Lite_A$, we have extended and used it as a part of our semantic privacy protection model. We have three layers of data sharing and integration infrastructure instead of two layers shown in $DL - Lite_A$ so we face a research challenge of ontology merging and rule integration from the middle layer to the top layer when we enforce a privacy protection policy (see Figure 3).

A semantic privacy protection model composed of three main components:

- In the top layer at the $\mathcal{VP}$, we have a global policy schema ($\mathcal{GPS}$), including a global ontology schema ($\mathcal{GS}$) aligned and merged from several local schemas ($\mathcal{LS}$), e.g. `TBox` and a set of rule integration at the middle layer. The $\mathcal{VP}$ provides conceptual data access and protection services that give users a unified conceptual "global view" with access control power for each data request.

- Ontology-based data sources are external, independent, and heterogeneous, and each local ontology was combined with logic program ($\mathcal{LP}$)-based rules for each server in the middle layer.

- Mapping language ($\mathcal{ML}$), which semantically links a $\mathcal{GS}$ and integrated rule set in the top layer to each server's ontology $\mathcal{LS}$ and privacy protection rules in the middle layer.

3. A FORMAL POLICY COMBINATION

A formal policy combination ($\mathcal{FPC}$) in a global policy schema ($\mathcal{GPS}$) allows data sharing as integration of $\mathcal{FP}$ from a variety of servers.

Each $\mathcal{FP}$ is shown as $\mathcal{K} = \mathcal{O} + \mathcal{R}$, where ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ and rule $\mathcal{R} = (\mathcal{RS}, \mathcal{F})$, $\mathcal{T}$ is `TBox`, and $\mathcal{A}$ is `ABox`; $\mathcal{RS}$ is a set of rules, and $\mathcal{F}$ is a set of facts.

$$\mathcal{FPC} = \underset{i}{\oplus}\mathcal{K}_i = (\underset{i}{\diamond}\mathcal{O}_i, \underset{i}{\odot}\mathcal{R}_i) = (\underset{i}{\diamond}(\mathcal{T}, \mathcal{A})_i, \underset{i}{\odot}(\mathcal{RS}, \mathcal{F})_i)$$
$$=((\underset{i}{\diamond}\mathcal{T}_i, \underset{i}{\diamond}\mathcal{A}_i), (\underset{i}{\odot}\mathcal{RS}_i, \underset{i}{\odot}\mathcal{F}_i))$$

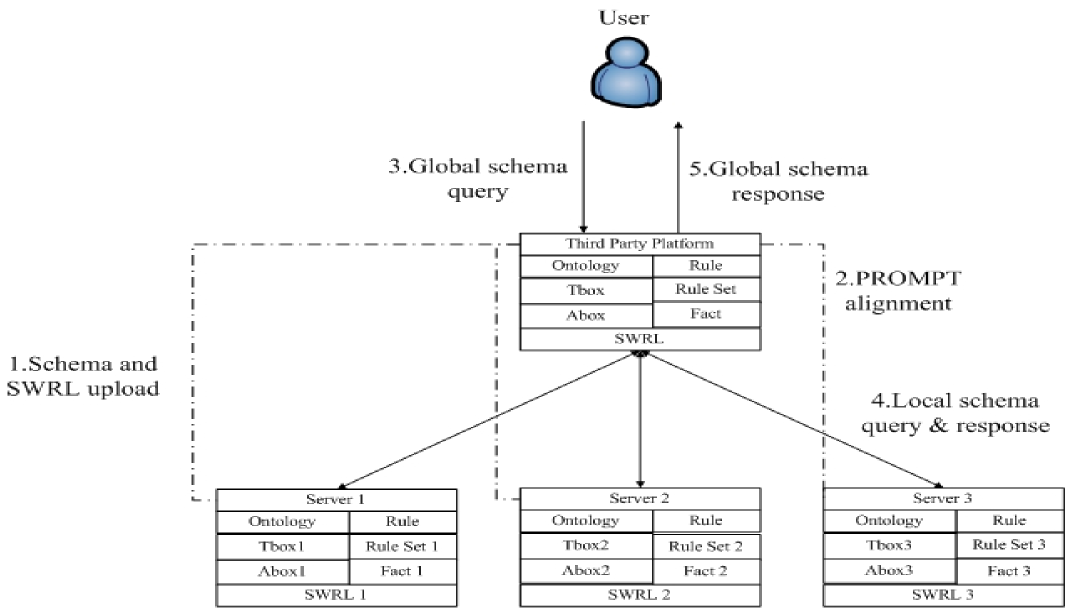

Figure 3: A virtual platform for ontology mapping, merging, and rule integration from multiple servers

where
i is the index of a server i.
$\oplus$ is an operator for formal policy combination,
$\diamond$ is an operator for ontology mapping and merging,
$\odot$ is an operator for rule integration.

In a semantic privacy protection model, a formal protection policy combination ($\mathcal{FPPC}$) allows data sharing and protection from $\mathcal{FPC} = \underset{i}{\oplus}\mathcal{K}_i = (\underset{i}{\diamond}\mathcal{O}_i, \underset{i}{\odot}\mathcal{R}_i)$, where $\underset{i}{\odot}\mathcal{R}_i = (\underset{i}{\odot}\mathcal{RS}_i, \underset{i}{\odot}\mathcal{F}_i)$ provides data query and protection services in $\underset{i}{\diamond}\mathcal{O}_i$.

3.1 $\mathcal{FPP}$ for Privacy Protection

A privacy protection policy is a type of $\mathcal{FPP}$. We designed an ontology that declares the FIPs' attributes as classes in an $\mathcal{FPP}$ (see Figure 4). The attributes, `purpose`, `datauser`, `data`, `obligation`, and `action` that allow people to specify the constraints of privacy protection policies using related property chains.

Constraint properties is a type of `owl` : `ObjectProperty` that specify what are the feasible domain and range classes of the above attributes. For example, a property `hasOptInPurpose` has its domain and range classes shown as follows:

$\mathtt{T} \sqsubseteq \forall$ `hasOptInPurpose.Data`,
$\mathtt{T} \sqsubseteq \forall$ `hasOptInPurpose`$^-$`.Purpose`.

Then a datalog rule, in the SWRL-based policy representation, allows us to use a property chain to combine the two feasible classes together:

$$\mathtt{hasOptInPurpose.Data}(?data)$$
$$\wedge\ \mathtt{hasOptInPurpose}^-\mathtt{.Purpose}(?purpose)$$
$$\longrightarrow \mathtt{hasOptInPurpose}(?data, ?purpose) \longleftarrow (1)$$

Similarly, a `hasOptInDatauser` property has its domain and range classes shown as follows:

$\mathtt{T} \sqsubseteq \forall$ `hasOptInDatauser.Data`,
$\mathtt{T} \sqsubseteq \forall$ `hasOptInDatauser`$^-$`.Datauser`.

Then another datalog rule allows us to use another property chain to combine another two feasible classes together:

$$\mathtt{hasOptInDatauser.Data}(?data)$$
$$\wedge\ \mathtt{hasOptInDatauser}^-\mathtt{.Datauser}(?datauser)$$
$$\longrightarrow \mathtt{hasOptInDatauser}(?data, ?datauser) \longleftarrow (2)$$

Based on (1) and (2), we have a feasible set of `ABox` instances with `data`, `purpose`, and `datauser` combinations of an attribute set that was permitted from the original `dataowner` to allow a particular type of `datauser` to ask for a `data` set with a permissive `purpose`. When a server collects a customer's data, the promise of data usage will be ensured if a data user's identity and usage purpose are verified successfully. Otherwise, the data will be kept secret without a data user's awareness.

These are easily extended to the other two attributes, `action` and `obligation`, to complete the FIPs' privacy protection criteria. An ordinary data user is allowed to ask a query service with `action` = `read` at the $\mathcal{VP}$. The other actions, such as `deletion` or `modify`, are only allowed for a system administrator in the middle layer when (s)he asks to delete a user's data to satisfy the obligation of data retention period or for a data owner updates his or her own profile data.

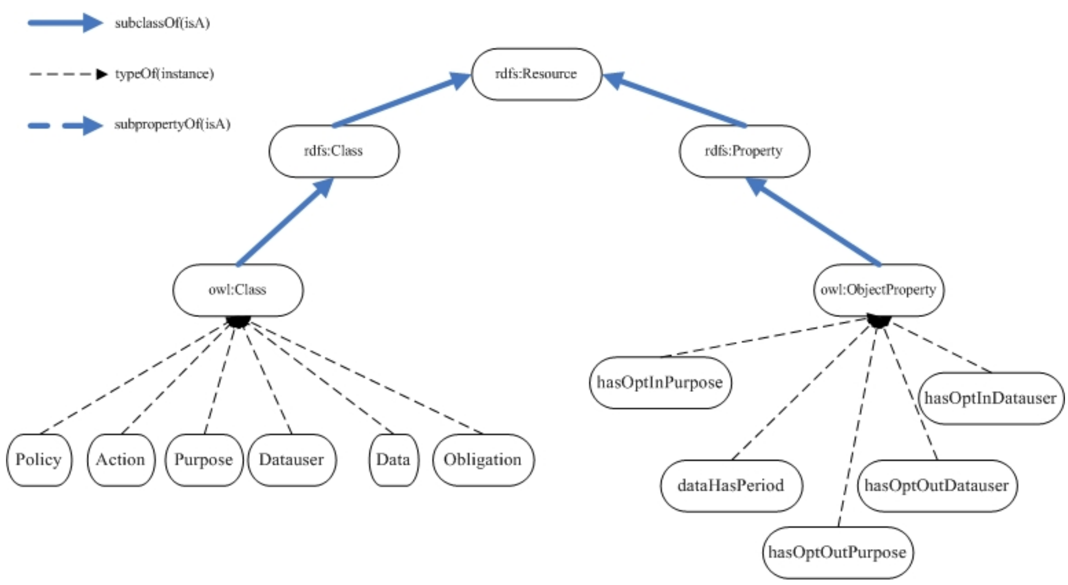

Figure 4: A partial ontology schema for OECD FIPs' attributes shown as $\mathtt{owl:Class}$, and constraints shown as $\mathtt{owl:Property}$

3.2 Data Request Services

A server declares its privacy policy in P3P before a data owner's data is collected. Once a user accepts a server's privacy declaration policy, the data usage constraints are specified as Figure 5, where FIP's five attributes $(?d, ?p, ?du, ?a, ?o)$ for `data`, `purpose`, `datauser`, `action`, and `obligation`, are classes, and `hasOptInDatauser`, `hasOptInPurpose`, etc., are properties proposed as chains of usage constraints for attributes. For each data request service, an initial feasible parameter input set is $\mathcal{FS} = input(?du, ?r, ?p)$, where $?du \in$ `Datauser`, $?r = read \in$ `Action`, $?p \in$ `Purpose` and output dataset with associated obligations is $output(?d, ?o)$, where $?d \in$ `Data`, $?o \in$ `Obligation`. The feasible dataset shown as `ABox` instances will be discovered by using SQWRL datalog rules. Further permissible actions will be activated when the following data protection policies are satisfied.

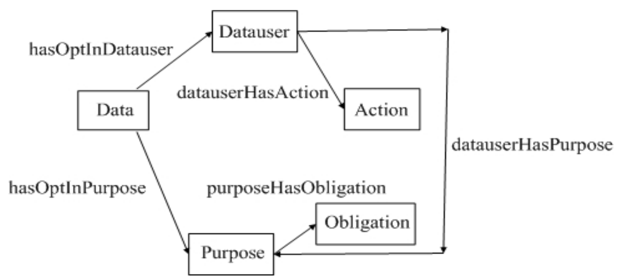

Figure 5: Five major FIP's attributes, such as data, purpose, etc are shown as $\mathtt{owl:class}$ and chained by associated $\mathtt{owl:Property}$, such as `hasOptInDatauser`, `hasOptInPurpose`, etc.

3.3 $\mathcal{FPPC}$ at the $\mathcal{VP}$

A data user still possibly collects a shareable data by asking each server individually without using a formal privacy protection policy combination ($\mathcal{FPPC}$). But the high complexity of using query services for all of data sources hinders people from using this data sharing approach. The other possible approach to collect a shareable data is to combine pair-wise servers' policies. Then, we face another scalability problem when more than two servers are intending to share their data.

In this semantic privacy-preserving model, we propose the $\mathcal{VP}$ infrastructure to allow a server in each data source to offer its $\mathcal{FPP}$ at the $\mathcal{VP}$ to enforce $\mathcal{FPPC}$. $\mathcal{FPP}$ in each data source is shown as $\mathcal{K} = \mathcal{O} + \mathcal{R}$, where ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ and rule $\mathcal{R} = (\mathcal{RS}, \mathcal{F})$. At the $\mathcal{VP}$, we only map and merge $\mathcal{T}$, e.g. `TBox` but leave $\mathcal{A}$, e.g. `ABox` instances in its original RDB data source. Similarly, we only integrate $\mathcal{RS}$, a set of rules at the $\mathcal{VP}$ but leave $\mathcal{F}$, a set of facts in its original RDB data source. The benefit of using this approach is to map and merge the `TBoxes` and to integrate the $\mathcal{RS}$ with the updated data only once.

4. ONTOLOGY MAPPING AND MERGING

A merged ontology come from mapping and alignment that provides data integration services. In particular, data integration through ontologies, such as LAV is possible for multiple servers if a mapping language $\mathcal{ML}$ provides a semantic mapping description between the $\mathcal{GS}$ and the underlying $\mathcal{LS}$ for each server [14]. In LAV, the relationships between the $\mathcal{GS}$ and the $\mathcal{LS}$ are established by making LAV assertions. Every assertions has the form $Q_{LS} \rightsquigarrow Q_{GS}$, where Q_{LS} provides the views of the conjunctive query ($\mathcal{CQ}$) over the global schema $\mathcal{GS}$ for each data source, and Q_{GS} is a $\mathcal{CQ}$ over the global schema $\mathcal{GS}$ at the $\mathcal{VP}$.

A $\mathcal{CQ}$ for Q_{LS} can be defined as a privacy-aware authorized view of each server so we do not disclose any non-shareable data to the $\mathcal{VP}$ whenever each server submits its $\mathcal{FPP}$ for ontology merging and rule integration. A $\mathcal{CQ}$ can be defined as a subset of Datalog program, i.e. $\mathcal{CQ}$ containment problem, for query the relational database. This problem was previously investigated in [34].

On the other hand, the connection between the problem of answering queries using extensions of views and the problem rewriting queries using views were studied before through an ontology expressed in DL [15]. In [8], a relational data integration was obtained by mapping each ontology element, e.g. class and property, in the $\mathcal{GS}$ into an SQL query of a relational data source. This is a GAV approach, focusing on mapping the elements of the $\mathcal{GS}$ to a view (SQL query) over the sources. However, our approach is more like LAV , where each term in a SQWRL query for each $\mathcal{LS}_i$ is defined as a view for a SQWRL query in the $\mathcal{GS}$.

4.1 Perfect Ontology Alignment

A mapping can be shown as (uid, e_1, e_2, n, ρ), where uid is a unique identity for the mapping, e_1, e_2 are entity names, such as class or property, and in the vocabulary of $\mathcal{O}_1$, $\mathcal{O}_2$, n is a numeric confidence measure between 0 and 1, and ρ is a relation such as subsumption ($\sqsubseteq$), equivalence ($\equiv$), or disjointness ($\perp$) between e_1 and e_2 [23].

In this study, the entity names for describing the ontology's class and property, and the structure of using these entity names in the root of the ontology schema for $\mathcal{O}_i$ to define the FIPs' privacy protection criteria (see Figure 5) that are required to be the same. This is a strict constraint to achieve a perfect ontology alignment of this study. Moreover, a perfect mapping language $\mathcal{ML}$ provides semantic mappings for each entity $e \in \mathcal{GS}$ at the $\mathcal{VP}$ to the corresponding entities $e_i \in \mathcal{LS}_i$.

A perfect ontology alignment obtained via a mapping (uid, e_i, e_j, n, ρ) and merging between $\mathcal{T}_i$ in $\mathcal{O}_i$ and $\mathcal{T}_j$ in $\mathcal{O}_j$ satisfies the following conditions:

- $e_i \in \mathcal{T}_i$ and $e_j \in \mathcal{T}_j$ entity names are either defined for describing the root class names which corresponding to the privacy protection concepts, such as `purpose`, `action`, `datauser`, `data`, and `obligation` or property names, such as `hasOptInDatauser`, `hasOptInPurpose`, etc; Furthermore entity names below the root class and root property are also defined for the descriptions of the underlying subclass and subproperty names.
- A numeric confidence measure n is always equal 1.
- ρ is either equivalence ($\equiv$) or subsumption ($\sqsubseteq$) between entity names of $\mathcal{T}_i$ and $\mathcal{T}_j$ schemas. In an equivalent ($\equiv$) case, we can find a pair of one-to-one corresponding entity names for $e_i \in \mathcal{T}_i$ and $e_j \in \mathcal{T}_j$ in the same layer of the respective ontology schema with $n = 1$; In a subsumption ($\sqsubseteq$) case, there are subclass or subproperty entity names not in the same layer so $e_i \in \mathcal{T}_i$ and $e_i \sqsubseteq e_j \in \mathcal{T}_j$, and vice versa.

4.2 Query Rewriting Services

SWRL combines OWL-DL's ontology language with an additional datalog rule language, where a datalog rule language is shown as an axiom of ontology, a little extension of the OWL-DL language that overcomes the limitations of property chaining in the OWL-DL language [20]. The computation complexity of answering SWRL-based policies might be undecidable regarding the verification of rights access permission unless these policies satisfy the $\mathtt{DL-Safe}$ conditions [29].

SPARQL is a query language for the RDF(S)-based ontologies. OWL2 QL is another query language for the OWL2-based ontologies. We did not use SPARQL query language or OWL2 QL, since our current local and global ontologies are modelled as the OWL-DL ontology language. In fact, SPARQL might not be able to query the complete semantics of the OWL-DL's ontologies. The OWL-DL's ontology queries can be shown as the SQWRL datalog rules, where the $\mathcal{CQ}$ conditions are shown as the rule's body and the query results, i.e., views are shown as the rule's conclusion. SQWRL uses SWRL's strong FOL semantic foundation as its formal semantics so this query language provides a small but powerful array of operators that allows users to construct queries over OWL-DL ontologies [31].

For each data request query service, a perfect mapping language $\mathcal{ML}$ provide the semantically linking of an entity name $e \in \mathcal{GS}$ in the datalog rule at the $\mathcal{VP}$ to the entity name $e_i \in \mathcal{LS}_i$ in the datalog rule at $server_i$, where $\mathcal{LS}_i$ is the `TBox` of $\mathcal{O}_i$, and e is a class or a property name. If there does not exist an $e_i \in \mathtt{TBox}_i$ in a subtree of the $\mathcal{LS}_i$ on the same layer as $e \in \mathtt{TBox}$ in the global tree of $\mathcal{GS}$, then we can recursively find a superclass or superproperty of e'_i with $e \sqsubseteq e'_i$ as the corresponding entity name, with a confidence measure of $n = 1$.

To successfully fulfill the semantically linking of any entity name $e \in \mathcal{GS}$ via $\mathcal{ML}$, an ontology schema designer must follow the principles we propose using the specifications of concepts and relations for the FIPs on the root layer of each ontology's local schema's $\mathcal{LS}_i$. But we still allow the designer to use different entity name string, $e_i \in \mathcal{LS}_i$ below the root layer of each local schema and to have an entirely different underlying subtree structure. We use *Prompt* ontology mapping algorithm first to synchronize the entity names between $\mathcal{LS}_i$ and further perform the ontology mappings and aligning operations. Finally we perfectly merge their schemas even if the subtrees of the local schemas are variant.

We use $\mathcal{ML}$ to map the name of a class entity $c \in \mathcal{GS}$ to one of the equivalent local ontology schema's class entity name in a deeper subtree, say $c_j \in \mathcal{LS}_j$, i.e., $c \leftrightsquigarrow c_j$ in the datalog rule's conditions of each data request service. When the class semantics for c is $c \sqsubseteq c_i$ in the $\mathcal{LS}_i$, i.e., we do not have a corresponding class $c'_i \in \mathcal{LS}_i$ on the same lower layer of a schema tree as $c \in \mathcal{GS}$. All of the `ABox` instances a_i in the class name entity c_i, i.e., $a_i \in c_i$ are still feasibly collected for this data request. Because class c_i is a legal domain class or range class for a particular property in the datalog rule for enforcing its privacy protection.

Similarly, a property $p \in \mathcal{GS}$ is mapped to another equivalent property $p_j \in \mathcal{LS}_j$ for the associated datalog rule's body conditions. Then property $p \leftrightsquigarrow p_j$ might be on a lower layer in the schema tree when compared with property $p_i \in \mathcal{LS}_i$. We still regard property p_i as feasible for its enforcement of the datalog rule on data sharing and protection. Finally, if we consider mappings for binding property and class from the aligning ontology schema $\mathcal{GS}$ to $\mathcal{LS}_i$ and $\mathcal{LS}_j$ to the respective datalog rule, then we have the following semantically linking relationships by using $\mathcal{ML}$'s mapping to align the ontology's class and property shown as follows:

Property $\mathtt{p} \in \mathcal{GS}$ with its domain class $\mathtt{dc}$ and range class $\mathtt{rc}$ that are mapped to property $\mathtt{p}_i \in \mathcal{LS}_i$ with its domain class $\mathtt{dc}_i$ and its range class $\mathtt{rc}_i$. For each data request service using a perfect mapping language $\mathcal{ML}$, when $\mathtt{p} \sqsubseteq \mathtt{p}_i$, we use property $\mathtt{p}_i$. Otherwise, when $\mathtt{p}_i \sqsubseteq \mathtt{p}$, we use property $\mathtt{p}$ for the datalog rule $\mathtt{r}_i$. When $\mathtt{dc} \sqsubseteq \mathtt{dc}_i$ and $\mathtt{rc} \sqsubseteq \mathtt{rc}_i$, we use class $\mathtt{dc}_i$ and $\mathtt{rc}_i$. Otherwise, when $\mathtt{dc}_i \sqsubseteq \mathtt{dc}$ and $\mathtt{rc}_i \sqsubseteq \mathtt{rc}$, we use class $\mathtt{dc}$ and $\mathtt{rc}$ for the datalog rule $\mathtt{r}_i$.

Here we did not explicitly consider an algebra operations, such as intersection or union, for `class/subclass` with property as shown in OWL-DL. Intuitively, this class/subclass and property algebra operation problem can be transformed to the generic class/property problem when terms from different data sources can be mapped and aligned at the $\mathcal{VP}$.

Example 1. In Figure 6, after we map and align two local partial ontology schemas, $\mathcal{LS}_1$ and $\mathcal{LS}_2$, into a merged partial ontology global schema $\mathcal{GS}$, we receive a data request service with class $\mathtt{P}_{212}$. In the purpose class $\mathtt{P}$, $\mathtt{P}_{111} \leftrightsquigarrow \mathtt{P}_{211}$, but $\mathtt{P}_{212} \in \mathcal{GS}$ does not have a corresponding subclass in $\mathcal{LS}_1$, since $\mathtt{P}_{212} \sqsubseteq \mathtt{P}_{21}$ and $\mathtt{P}_{21} \leftrightsquigarrow \mathtt{P}_{11}$. When a data request service asks for class $\mathtt{P}_{212} \in \mathcal{GS}$, mapping language $\mathcal{ML}$ will map $\mathtt{P}_{212}$ to $\mathtt{P}_{11}$ for the datalog rule $\mathtt{r}_i$ to query the $\mathcal{LS}_1$.

5. PERFECT RULE INTEGRATION

In $\mathcal{FPPC}$, we define an integrated rule set $\underset{i}{\odot}\mathcal{R}_i = (\underset{i}{\odot}\mathcal{RS}_i, \underset{i}{\odot}\mathcal{F}_i)$ to enforce data query and protection services in $\underset{i}{\diamond}\mathcal{O}_i$. In fact, an integrated rule set $\underset{i}{\odot}\mathcal{RS}_i$ is a part of $\mathcal{FPC}$ that was created by collecting the datalog rules, e.g. SQWRL queries, in the formal policies $\mathcal{FP}_i$, from local servers. A datalog rule $\mathtt{r}_i$ in the $\mathcal{R}_i$ of $\mathcal{FP}_i$ is shown as [2]:

$$\mathcal{H} \longleftarrow \mathcal{B}_1 \wedge \mathcal{B}_2 \wedge, \cdots, \wedge \mathcal{B}_n,$$

where $\mathcal{H}$, the query results (or views) are expressed as SQWRL built-ins, such as $\mathtt{sqwrl:select}$ and the rule antecedent $\mathcal{B}_i$, are defined as a pattern matching specifications, i.e., query conditions that are either SQWRL built-ins or class and property predicates from the ontology schema.

A perfect rule integration is defined for the integration of any datalog rules as: $\exists \mathtt{r}_i \in \mathcal{RS}_i$ in $\mathcal{FP}_i$, for the purpose of data sharing and protection without causing conflicts with $\exists \mathtt{r'}_i \in \underset{i}{\odot}\mathcal{R}_i$, $\lambda_i \in \underset{i}{\diamond}\mathcal{O}_i$, i.e., conditions do not exist for $\exists \mathtt{r}_i \models \lambda_i \Rightarrow \exists \mathtt{r'}_i \not\models \lambda_i$, or $\exists \mathtt{r}_i \not\models \lambda_i \Rightarrow \exists \mathtt{r'}_i \models \lambda_i$. Then, $\exists \mathtt{r'}_i \in \underset{i}{\odot}\mathcal{R}_i$ at the $\mathcal{VP}$ can be activated and mapped by the perfect mapping language $\mathcal{ML}$ into $\exists \mathtt{r}_i$, to enable a global data query and protection service of multiple servers.

[2]This datalog rule is related to a $\mathcal{CQ}$ of the form: $v_i \leftarrow conj_i(\overrightarrow{x}_i)$ [9]

Example 2. A rule $\mathtt{r'}_i$ is one of the rules within the integrated rule set at the $\mathcal{VP}$. It asks for a data set ?d with related obligations ?o under the feasible parameter input set $\mathcal{FS}_i = (\mathtt{M1}, \mathtt{TMarketing6}, \mathtt{Read2})$, where data user M1 is a marketing staff with a purpose of achieving telephone marking TMarketing, A rule $\mathtt{r'}_i$ is mapped to a rule $\mathtt{r}_i$ and a rule $\mathtt{r}_j$ using the rule mapping processes when we have done an upward perfect ontology mapping, alignment, merging and a perfect rule integration. A downward perfect mapping language $\mathcal{ML}$ operation maps the $\mathtt{r'}_i$'s predicates, such as class, property to the corresponding predicates in a rule $\mathtt{r}_i$ and a rule $\mathtt{r}_j$ with $\mathtt{MUser(M1)} \sqsubseteq \mathtt{Datauser(M1)}$, $\mathtt{TMarketing(TMarketing6)} \sqsubseteq \mathtt{Purpose(TMarketing6)}$. Therefore, real data query and protection services requested by a rule $\mathtt{r'}_i$ are performed by a rule $\mathtt{r}_i$ and a rule $\mathtt{r}_j$.

A rule $\mathtt{r'}_i$ queries at the $\forall i \ \underset{i}{\diamond}\mathcal{O}_i$:
$\underline{\mathtt{MUser(M1)}} \wedge \underline{\mathtt{TMarketing(TMarketing6)}}$
$\wedge \mathtt{datauserHasPurpose(M1, TMarketing6)}$
$\wedge \mathtt{datauserHasAction(M1, Read2)}$
$\wedge\ \mathtt{hasOptInPurpose(?d, TMarketing6)}$
$\wedge \mathtt{hasOptInDataUser(?d, M1)}$
$\wedge\ \mathtt{purposeHasObligation(TMarketing6, ?o)}$
$\longrightarrow \mathtt{sqwrl:selectDistinct(?d, M1, TMarketing6, Read2, ?o)}$

A rule $\mathtt{r}_i$ queries at the $\mathcal{O}_i$:
$\underline{View(\mathtt{Datauser(M1)})} \wedge \underline{View(\mathtt{TMarketing(TMarketing6)})}$
$\wedge\ \mathtt{datauserHasPurpose(M1, TMarketing6)}$
$\wedge\ \mathtt{datauserHasAction(M1, Read2)}$
$\wedge\ \mathtt{hasOptInPurpose(?d, TMarketing6)}$
$\wedge\ \mathtt{hasOptInDataUser(?d, M1)}$
$\wedge\ \mathtt{purposeHasObligation(TMarketing6, ?o)}$
$\longrightarrow \mathtt{sqwrl:selectDistinct(?d, M1, TMarketing6, Read2, ?o)}$

A rule $\mathtt{r}_j$ queries at the $\mathcal{O}_j$:
$\underline{View(\mathtt{MUser(M1)})} \wedge \underline{View(\mathtt{Purpose(TMarketing6)})}$
$\wedge\ \mathtt{datauserHasPurpose(M1, TMarketing6)}$
$\wedge\ \mathtt{datauserHasAction(M1, Read2)}$
$\wedge\ \mathtt{hasOptInPurpose(?d, TMarketing6)}$
$\wedge\ \mathtt{hasOptInDataUser(?d, M1)}$
$\wedge\ \mathtt{purposeHasObligation(TMarketing6, ?o)}$
$\longrightarrow \mathtt{sqwrl:selectDistinct(?d, M1, TMarketing6, Read2, ?o)}$

Example 3. Under the data protection law, two hospitals, A and B, have allowed to share their patients' Electronic Health Records (EHRs) after patients give their consents for the medication purpose . A patient was hospitalized in the hospital A for a surgery. After that, this patient went to the hospital B for an outpatient medication. A physician in the hospital B was authorized to query this patient's shareable EHR at the $\mathcal{VP}$ collected from hospital A and hospital B's RDB data sources. The vocabularies of partial ontology schemas for hospital A's local schema LS_A, hospital B's local schema LS_B, and the global schema GS at the $\mathcal{VP}$ are shown as Figure 7.

Hospital A has the following terms as its ontology's local schema LS_A vocabularies:
`Class`: `Clinic` and `HealthData` with subClass `SurgeryData` and `HospitalizationData`
`Property`: `create` with domain class as `Clinic` and range

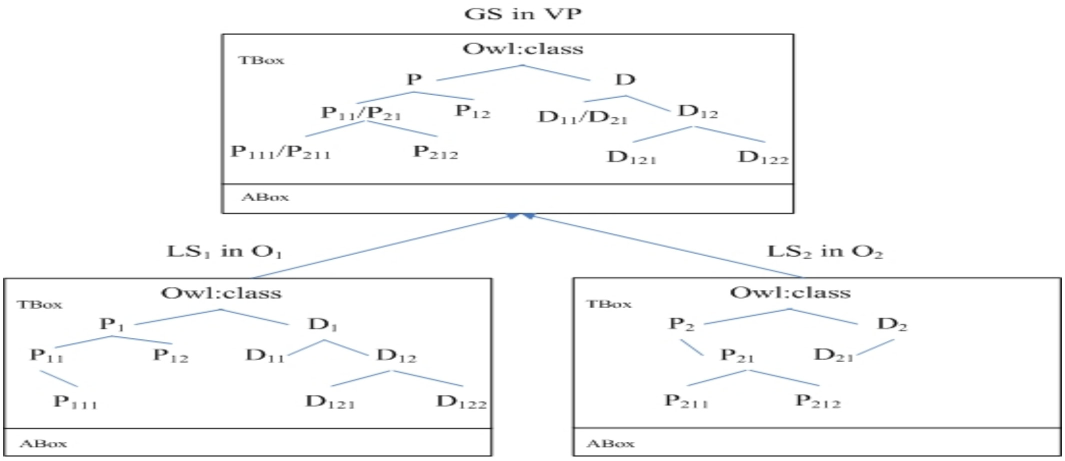

Figure 6: Partial ontology mapping for class alignment and ontology merging

class as `HealthData`, i.e.,
$\mathtt{T} \sqsubseteq \forall$ `create.Clinic`
$\mathtt{T} \sqsubseteq \forall$ $\mathtt{create}^{-}$`.HealthData`

Hospital `B` has the following terms as its ontology's local schema LS_B vocabularies:
`Class: Person, HealthCenter,` and `PatientData` with sub-Class `OutPatientData`
`Property: own, beMedicared` with their respective domain and range class are shown as follows:

$\mathtt{T} \sqsubseteq \forall$ `own.Person`, $\mathtt{T} \sqsubseteq \forall$ $\mathtt{Own}^{-}$`.PatientData`.
$\mathtt{T} \sqsubseteq \forall$ `beMedicated.Person`,
$\mathtt{T} \sqsubseteq \forall$ $\mathtt{beMedicated}^{-}$`.HealthCenter`.

The $\mathcal{VP}$ offers the following vocabularies:
`Class: Patient, Hospital, Surgery,` and `HealthRecord`
`Property: beCured, hasHealthRecord, generate` with their respective domain and range class are shown as follows:

$\mathtt{T} \sqsubseteq \forall$ `beCured.Patient`, $\mathtt{T} \sqsubseteq \forall$ $\mathtt{beCured}^{-}$`.Hospital`
$\mathtt{T} \sqsubseteq \forall$ `hasHealthRecord.Patient`
$\mathtt{T} \sqsubseteq \forall$ $\mathtt{hasHealthRecord}^{-}$`.HealthRecord`
$\mathtt{T} \sqsubseteq \forall$ `generate.Hospital`
$\mathtt{T} \sqsubseteq \forall$ $\mathtt{generate}^{-}$`.HealthRecord`

Use LAV approach to define each class and property in these two hospital local schemas as views in terms of the global schema's vocabularies shown as follows:

Views use at the $\mathcal{VP}$ created from the hospital `A` local schema's vocabularies are:

$\mathtt{def}(\mathtt{V1}_{\mathtt{Clinic}}) = \mathtt{Hospital}$
$\mathtt{def}(\mathtt{V2}_{\mathtt{HealthData}}) = \mathtt{HealthRecord}$
$\mathtt{def}(\mathtt{V3}_{\mathtt{SuregeryData}})$
$= \mathtt{HealthRecord} \wedge \forall \mathtt{hasMedType.Surgery}$

$\mathtt{def}(\mathtt{V4}_{\mathtt{HospitalizationData}})$
$= \mathtt{HealthRecord} \wedge \forall\ \mathtt{hasMedType.Hospitalization}$
$\mathtt{def}(\mathtt{V5}_{\mathtt{create}}) = \mathtt{generate}$

Views use at the $\mathcal{VP}$ created from the hospital `B` local schema's vocabularies are:

$\mathtt{def}(\mathtt{V6}_{\mathtt{Person}}) = \mathtt{Patient}$
$\mathtt{def}(\mathtt{V7}_{\mathtt{HealthCenter}}) = \mathtt{Hospital}$
$\mathtt{def}(\mathtt{V8}_{\mathtt{PatientData}}) = \mathtt{HealthRecord}$
$\mathtt{def}(\mathtt{V9}_{\mathtt{OutPatientData}})$
$= \mathtt{HealthRecord} \wedge \forall\ \mathtt{hasMedType.OutPatient}$
$\mathtt{def}(\mathtt{V10}_{\mathtt{beMedicated}}) = \mathtt{beCured}$
$\mathtt{def}(\mathtt{V11}_{\mathtt{own}}) = \mathtt{hasHealthRecrod}$

A physician queries a patient's surgery record at the $\mathcal{VP}$ by using a merged global ontology schema based on LAV query rewriting instead of directly requesting each hospital. An original datalog-based SQWRL rule for a query `q` at the $\mathcal{VP}$ is shown as:

$\mathtt{Patient}(?\mathtt{x}) \wedge \mathtt{beCured}(?\mathtt{x}, ?\mathtt{y}) \wedge \mathtt{hasHealthRecrod}(?\mathtt{x}, ?\mathtt{r})$
$\wedge\ \mathtt{HealthRecord}(?\mathtt{r}) \wedge \mathtt{hasMedType}(?\mathtt{r}, \mathtt{Surgery})$
$\wedge\ \mathtt{generate}(?\mathtt{y}, ?\mathtt{r}) \longrightarrow \mathtt{sqwrl} : \mathtt{select}(?\mathtt{x}, ?\mathtt{r})$

Query rewriting of the `q` in terms of two $\mathcal{CQ}$s, e.g., $\mathtt{q}_{\mathtt{va}}$ and $\mathtt{q}_{\mathtt{vb}}$, uses views defined at the $\mathcal{VP}$:

$\mathtt{V6}_{\mathtt{Person}} \wedge \mathtt{V10}_{\mathtt{beMedicated}} \wedge \mathtt{V11}_{\mathtt{own}} \wedge \mathtt{V9}_{\mathtt{OutPatientData}} \wedge \mathtt{V5}_{\mathtt{create}}$
$\longrightarrow \mathtt{sqwrl} : \mathtt{select}(?\mathtt{x}, ?\mathtt{r}) \longleftarrow (\mathtt{q}_{\mathtt{va}})$

Above $\mathtt{q}_{\mathtt{va}}$ query is corresponding to a query as:
$\mathtt{B} : \mathtt{Person}(?\mathtt{p}) \wedge \mathtt{B} : \mathtt{beMedicated}(?\mathtt{p}, ?\mathtt{c}) \wedge \mathtt{B} : \mathtt{own}(?\mathtt{p}, ?\mathtt{d})$
$\wedge\ \mathtt{B} : \mathtt{OutPatientData}(?\mathtt{od}) \wedge \mathtt{A} : \mathtt{create}(?\mathtt{h}, ?\mathtt{hd})$
$\longrightarrow \mathtt{sqwrl} : \mathtt{select}(?\mathtt{p}, ?\mathtt{od})$

$\mathtt{V6}_{\mathtt{Person}} \wedge \mathtt{V10}_{\mathtt{beMedicated}} \wedge \mathtt{V11}_{\mathtt{own}} \wedge \mathtt{V3}_{\mathtt{SuregeryData}} \wedge \mathtt{V5}_{\mathtt{create}}$
$\longrightarrow \mathtt{sqwrl} : \mathtt{select}(?\mathtt{x}, ?\mathtt{r}) \longleftarrow (\mathtt{q}_{\mathtt{vb}})$

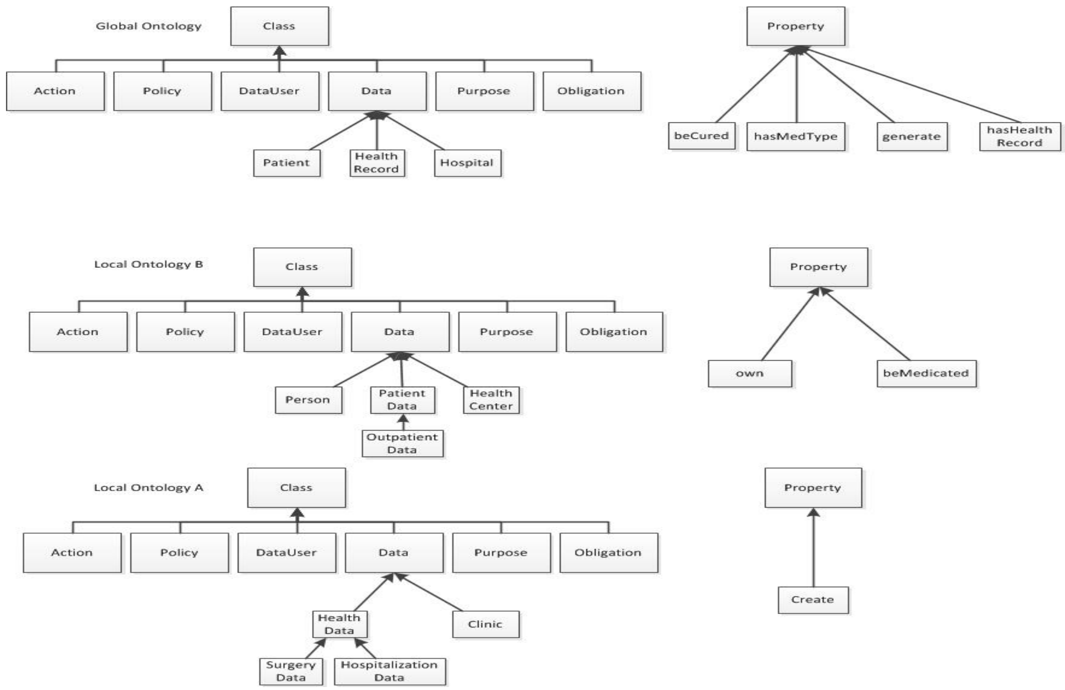

Figure 7: A partial ontology for Electronic Health Record (EHR) sharing and privacy protection

Above $\mathtt{q_{vb}}$ query is corresponding to a query as:
$\mathtt{B}:\mathtt{Person}(?\mathtt{p}) \wedge \mathtt{B}:\mathtt{beMedicated}(?\mathtt{p},?\mathtt{c}) \wedge \mathtt{B}:\mathtt{own}(?\mathtt{p},?\mathtt{d})$
$\wedge\ \mathtt{A}:\mathtt{SuregeryData}(?\mathtt{sd}) \wedge \mathtt{A}:\mathtt{create}(?\mathtt{h},?\mathtt{hd})$
$\longrightarrow \mathtt{sqwrl}:\mathtt{select}(?\mathtt{p},?\mathtt{sd})$

6. SOUNDNESS AND COMPLETENESS

In this section, we briefly demonstrate how the exact query rewriting service satisfies the soundness and completeness criteria by using the LAV source descriptions based on the $\mathcal{GPS} = (\underset{i}{\diamond}\mathcal{O}_i, \underset{i}{\odot}\mathcal{R}_i)$ at the $\mathcal{VP}$: If $\mathtt{q}(\mathtt{x})$ is a $\mathcal{CQ}$ over $\underset{i}{\diamond}\mathcal{O}_i$ at the $\mathcal{VP}$ and $\mathtt{q_{vi}}(\mathtt{x})$ is a $\mathcal{CQ}$ over $\mathcal{O}_i$ using LAV source descriptions from a data $server_i$, then $\forall x \ \ q(x) \longleftrightarrow \bigsqcup_i q_{vi}(x)$. In [15], authors showed that when a query has a finite number of *maximally contained conjunctive rewritings*, then the complete set of its answers can be obtained as the union of the answer sets of its rewritings. The *datalog-rewriting* was introduced, in which query language is a *hybrid language* with CARIN as its combination of $\mathcal{O}+\mathcal{R}$, and the rewriting language is a *relational language*. They also provided a rewriting algorithm, and showed that the *RewriteQuery* is sound and complete.

In comparison, we use LAV for rewriting queries and use SWRL as a combination of $\mathcal{O}+\mathcal{R}$. A perfect ontology merging and a rule integration ensure the soundness and completeness of data sharing and integration in the semantic privacy-preserving model. This will be briefly shown as follow:

6.1 [Soundness]

For the *soundness* criterion, we do not allow any unintentionally released (or protected) data for a user by using a query rewriting service with a rule (query) $\mathtt{r}'_i \in \underset{i}{\odot}\mathcal{R}_i$ at the $\mathcal{VP}$ instead of using a direct query service as rules (queries) $\mathtt{r}_i \in \mathcal{R}_i$ in each $server_i$, $\forall i$.

THEOREM 1. **[Soundness]** *After a perfect ontology alignment and rule integration with $\mathcal{FPPC}$, $\exists \mathcal{GPS} = (\underset{i}{\diamond}\mathcal{O}_i, \underset{i}{\odot}\mathcal{R}_i)$ at the $\mathcal{VP}$, Under a particular feasible parameter input set $\mathcal{FS}_i$, if $\lambda_j \in \mathcal{O}_i$ is protected by a $\mathcal{FPP}_i$ at each $server_i, \forall i$, i.e., $\forall i, r_i \in \mathcal{R}_i \nvDash \lambda_j$, then $\mathtt{r}'_i \in \underset{i}{\odot}\mathcal{R}_i \nvDash \lambda_j$ for the same $\mathcal{FS}_i$, where λ_j is a protective data set in $\mathcal{O}_i$.*

PROOF. (Sketch) If $\mathtt{q}(\mathtt{x})$ is a query over $\underset{i}{\diamond}\mathcal{O}_i$ at the $\mathcal{VP}$ and $\mathtt{q_{vi}}(\mathtt{x})$ is a query over $\mathcal{O}_i$ in a $server_i$, then we need to prove the statement $\forall x \ \ q(x) \longrightarrow \bigsqcup_i q_{vi}(x)$. This statement is equivalent to the original argument: If $r_i \in \mathcal{R}_i \nvDash \lambda_j$, then $\mathtt{r}'_i \in \underset{i}{\odot}\mathcal{R}_i \nvDash \lambda_j$. The $\mathcal{CQ}$ $\mathtt{q}(\mathtt{x})$ is a query containment of datalog rule $\mathtt{r}'_i$ and the $\mathcal{CQ}$ $\mathtt{q_{vi}}(\mathtt{x})$ is a query containment of datalog rule $r_i \in \mathcal{R}_i$. The statement $\forall x \ \ q(x) \longrightarrow \bigsqcup_i q_{vi}(x)$ is true because the local as view (LAV) schema mapping only allow the protected concept λ_j in each $server_i$ to be connected to the global schema. After using a perfect ontology alignment and a perfect rule integration with a perfect mapping language $\mathcal{ML}$, we avoid the following condition: $\exists \mathtt{r}_i \nvDash \lambda_j \Rightarrow \exists \mathtt{r}'_i \models \lambda_j$. $\square$

6.2 [Completeness]

As for the *completeness* criterion, we do not allow any eligible shared data being missed for a query by a query rewriting service with a rule (query) $\mathtt{r}'_i \in \underset{i}{\odot}\mathcal{R}_i$ at the $\mathcal{VP}$ instead of using a direct query service as rules (queries) $\mathtt{r}_i \in \mathcal{R}_i$ in each $server_i$, $\forall i$.

THEOREM 2. **[Completeness]** *After a perfect ontology alignment and rule integration with $\mathcal{FPPC}$, $\exists \mathcal{GPS} = (\underset{i}{\diamond}\mathcal{O}_i, \underset{i}{\odot}\mathcal{R}_i)$ at the $\mathcal{VP}$, Under a particular feasible parameter input set $\mathcal{FS}_i$, if $\lambda_j \in \mathcal{O}_i$ is shareable by a $\mathcal{FPP}_i$ at each $server_i, \forall i$, i.e., $\forall i, r_i \in \mathcal{R}_i \models \lambda_j$, then $\mathtt{r'}_i \in \underset{i}{\odot}\mathcal{R}_i \models \lambda_j$ for the same $\mathcal{FS}_i$, where λ_j is a shareable data set in $\mathcal{O}_i$.*

PROOF. (Sketch) If $\mathtt{q}(\mathtt{x})$ is a query over $\underset{i}{\diamond}\mathcal{O}_i$ at the $\mathcal{VP}$ and $\mathtt{q_{vi}}(\mathtt{x})$ is a query over $\mathcal{O}_i$ in a $server_i$, then we need to prove the statement $\forall x \;\; q(x) \longleftarrow \bigsqcup_i q_{vi}(x)$. This statement is equivalent to the original argument: If $r_i \in \mathcal{R}_i \models \lambda_j$, then $\mathtt{r'}_i \in \underset{i}{\odot}\mathcal{R}_i \models \lambda_j$. The $\mathcal{CQ}$ $\mathtt{q}(\mathtt{x})$ is a query containment of datalog rule $\mathtt{r'}_i$ and the $\mathcal{CQ}$ $\mathtt{q_{vi}}(\mathtt{x})$ is a query containment of datalog rule $r_i \in \mathcal{R}_i$. The statement $\forall x \;\; q(x) \longleftarrow \bigsqcup_i q_{vi}(x)$ is true because the local as view (LAV) schema mapping only allows all of the shareable concepts λ_j in each $server_i$ to be exported to the global schema. After using a perfect ontology alignment method and a perfect rule integration method with a perfect mapping language $\mathcal{ML}$, we avoid the following condition: $\exists \mathtt{r}_i \models \lambda_j \Rightarrow \exists \mathtt{r'}_i \not\models \lambda_j$. $\square$

7. RELATED WORK

Data integration is a pervasive challenge faced in the applications that need to query across multiple autonomous and heterogeneous data sources. This problem has been received considerable attention from researchers in the fields of Artificial Intelligence and Database System more than a decade [18] [27]. A logic of the Description Logic (DL) family is used to model the ontology managed by the integration system, to formulate queries posed to the system, and to perform several types of automated reasoning supporting both the modeling, and the query answering process [11]. The ontology expresses the domain of interest of the information system at a high level of abstraction, and the relationship between data at the sources and instances of concepts and roles in the ontology is expressed by means of mappings, such as GLAV, GAV, LAV [7] [33].

Unfortunately, data integration and sharing are hampered by legitimate and widespread privacy concerns so it is critical to develop a technique to enable the integration and sharing of data without losing privacy. We face a challenge to develop a privacy framework for data integration that is flexible and clear to the end users [12]. View-based query answering over DL provides a framework to answer a query under the assumption that the only accessible information consists of the precomputed answers to a set of queries, called views. Privacy-aware access to data, each user is associated with a set of views, called authorization views, which specify the information that the user is allowed to access [9].

We encompass and extend previous ontology-based data integration system. A semantic privacy-preserving model provides authorized view-based query answering over a widespread multiple servers for data sharing and integration. The combined semantics-enabled privacy protection policies are used to empower the data integration and access control services at the virtual platform.

The role-based access control (RBAC) model is used to enforce the access control policies with a static role assignment for a stand-alone system. It is therefore not useful for solving the privacy protection problem. In fact, the RBAC model did not consider the prime elements of the FIPs, so it is not intended for a privacy protection problem. In [32], the $UCON_{ABC}$ might be useful for the privacy protection problem, but it did not explicitly allow the data sharing and protection in multiple sites.

The EFAF access control model is an extension of the FAF that provided the solution for privacy protection [22] [24]. This is the closest method to our solution, but its privacy protection control is more on the logic program and less on the ontology schema for the structure data modelling. This also prevents the data sharing and protection in multiple sites. The other similar models for enforcing the enterprise privacy protection go to the following EPAL [25] [35]. OASIS XACML is a policy language for privacy and digital rights protection. But it is an XML-based policy language so the policies based on XACML possibly might have ambiguous semantics that prevent using a flexible policy combination in multiple servers [1].

8. CONCLUSION AND FURTHER STUDY

We propose a semantic privacy protection model which encompasses and extends the existing works on data sharing and integration. We intend to solve the privacy protection problem to provide data sharing and integration in multiple servers by using one of ontology and rule language combinations, e.g. SWRL. Another OWL2 combination will be considered for the future [17]. This can be extended to a modular reuse of ontologies for data sharing and protection in the cross-domain cloud computing environment [16].

The perfect ontology alignment through ontology mapping and merging creates a global ontology schema at the $\mathcal{VP}$ by integrating multiple local ontology schemas from different data sources. In addition, the perfect rule integration by the perfect mapping language avoids any possible data usage conflicts between datalog rules from different data sources at the $\mathcal{VP}$. In fact, a datalog rule is considered as a conjunctive query, which provides data query and protection services in each server.

However this perfect ontology alignment is impossible without the requirements of using same ontology schema for the root layers for multiple servers with the LAV schema mapping. We face another policy hidden conflict challenge if background default policy assumptions are vary between different servers. All of these need further study.

Finally semantics-enabled policies are combined together at the $\mathcal{VP}$, so we simplify the data sharing and protection services. But the soundness and completeness criteria are still preserved for data sharing and integration purposes. This supports the trustworthiness of a policy combination for multiple servers.

Acknowledgements

This research was partially supported by the NSC Taiwan under Grant No. NSC 99-2221-E-004-010.

9. REFERENCES

[1] A. H. Anderson. A comparison of two privacy policy languages: EPAL and XACML. In *Proceedings of the 3rd ACM Workshop on Secure Web Services (SWS'06)*, pages 53–60. ACM, 2006.

[2] I. A. Antón et al. A roadmap for comprehensive online for privacy policy management. *Comm. of the ACM*, 50(7):109–116, July 2007.

[3] A. P. Bernstein and L. M. Haas. Information integration in the enterprise. *Comm. of the ACM*, 51(8):72–79, July 2008.

[4] A. P. Bonatti et al. An algebra for composing access control policies. *ACM Trans. on Information and Systems Security*, 5(1):1–35, February 2002.

[5] P. Bonatti and D. Olmedilla. Policy language specification, enforcement, and integration. project deliverable D2, working group I2. Technical report, REWERSE, 2005.

[6] J. d. Bruijn. RIF RDF and OWL compatibility. Technical report, W3C, Oct. 2009.

[7] D. Calvanese et al. Description logic framework for information integration. In *Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning*, pages 2–13. Morgan Kaufmann, 1998.

[8] D. Calvanese et al. Data integration through $DL-Lite_A$ ontologies. In *3rd Int. Workshop on Semantics in Data and Knowledge Base (SDKB)*, volume 4925, pages 26–47. Springer, 2008.

[9] D. Calvanese et al. View-based query answering over description logic ontologies. In *Proc. of KR-2008*. AAAI Press, 2008.

[10] D. Calvanese and G. D. Giacomo. Data integration: A logic-based perspective. *AI Magazine*, 26(1):59–70, 2005.

[11] D. Calvanses et al. Description logics for information integration. In *Computational Logic*, LNAI 2408, pages 41–60. Springer, 2002.

[12] C. Clifton et al. Privacy-preserving data integration and sharing. In *Data Mining and Knowledge Discovery*, pages 19–26. ACM, 2004.

[13] J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer-Verlag, 2007.

[14] M. Friedman et al. Navigational plans for data integration. In *Proc. of the Sixteen National Conference on Artificial Intelligence (AAAI'99)*, pages 67–73. AAAI/MIT Press, 1999.

[15] F. Goasdoué and M.-C. Rousset. Answering queries using views: a KRDB perspective for the semantic web. *ACM Trans. on Internet Technology*, 4(3):255–288, August 2004.

[16] C. B. Grau et al. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research*, pages 273–318, 2008.

[17] C. B. Grau et al. OWL2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web 3*, pages 309–322, 2008.

[18] A. Halevy, A. Rajaraman, and J. Ordille. Data integration: The teenage years. In *VLDB'06*, pages 9–16. ACM, 2006.

[19] Y. A. Halevy. Answering queries using views: A survey. *The VLDB Journal*, 10(4):270–294, 2001.

[20] I. Horrocks et al. OWL rules: A proposal and prototype implementation. *Web Semantics: Science, Services and Agents on the World Wide Web 3*, 3(1):23–40, 2005.

[21] Y. J. Hu and H. Boley. Sempif: A semantic meta-policy interchange format for multiple web policies. In *2010 IEEE/WIC/ACM Int. Conference on Web Intelligence and Intelligent Agent Technology*, pages 302–307. IEEE, 2010.

[22] S. Jajodia et al. Flexible support for multiple access control policies. *ACM Trans. on Database Systems*, 26(2):214–260, June 2001.

[23] E. Jiménez-Ruiz et al. Ontology integration using mappings: Towards getting the right logical consequences. In *ESWC 2009*, LNCS 5554, pages 173–187. Springer, 2009.

[24] G. Karjoth and M. Schunter. A privacy policy model for enterprises. In *15th IEEE Computer Security Foundations Workshop (CSFW)*. IEEE, June 2002.

[25] G. Karjoth, M. Schunter, and E. V. Herreweghen. Translating privacy practices into privacy promises - how to promise what you can keep. In *POLICY'03*. IEEE, 2003.

[26] M. Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, pages 233–246. ACM, 2002.

[27] Y. A. Levy. Logic-based techniques in data integration. In T. Yu and S. Jajodia, editors, *Logic-based Artificial Intelligence*, pages 1–27. Kulwer, 2001.

[28] P. Mazzoleni et al. XACML policy integration algorithms. *ACM Trans. on Information and System Security*, 11(1), 2008.

[29] B. Motik, U. Sattler, and R. Studer. Query answering for OWL-DL with rules. In *3rd International Semantic Web Conference (ISWC) 2004*, LNCS 3298, pages 549–563. Springer, 2004.

[30] A. Nash and A. Deutsch. Privacy in GLAV information integration. In *ICDT 2007*, LNCS 4353, pages 89–103. Springer, 2007.

[31] J. M. O'Connor and A. K. Das. SQWRL: a query language for OWL. In *OWLED*, volume 529. CEUR, 2009.

[32] J. Park and R. T. Sandhu. The UCON_{ABC} usage control model. *ACM Trans. on Information and System Security*, 7(1):128–174, 2004.

[33] A. Poggi et al. Linking data to ontologies. *Journal on Data Semantics X*, 4900:133–173, 2008.

[34] D. J. Ullman. Information integration using logical views. *Theoretical Computer Science*, 239:189–210, 2000.

[35] S. D. C. d. Vimercati et al. Access control policies and languages in open environments. In T. Yu and S. Jajodia, editors, *Secure Data Management in Decentralized Systems*, pages 21–58. Springer, 2007.