Towards Law-Aware Semantic Cloud Policies with Exceptions for Data Integration and Protection*

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

Win-Nan Wu
ENT Lab., Dept. of CS
National Chengchi University
Taipei, Taiwan, 11605
99753505@nccu.edu.tw

Di-Rong Cheng
ENT Lab., Dept. of CS
National Chengchi University
Taipei, Taiwan, 11605
long@nccu.edu.tw

ABSTRACT

The main issues related to cloud computing implementation are security, privacy, and law-awareness. We consider data protection with law-awareness as the major concern for cloud service providers (CSPs) and their customers. Therefore, we provide Law-as-a-Service (LaaS) for CSPs on our law-aware semantic cloud policy infrastructure. The semantic legal policies in compliance with the laws are enforced automatically at the super-peer to enable LaaS. This allows CSPs to deploy their cloud resources and services without worrying about law violations. Afterward, users could query data from the law-aware super-peer within a super-peer domain. Each query is also compliant with the laws. Policies are shown as a combination of OWL-DL ontologies and stratified Datalog rules with negation for a policy's exceptions handling through defeasible (or non-monotonic) reasoning. Finally, the proof-of-concepts prototype systems have been implemented for an H1N1 pandemic investigation scenario in the semantic cloud to justify our approach.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*data integration*; K.4.1 [**Computers and Society**]: Public Policy Issues—*privacy, national security, information and laws*

General Terms

WWW, Semantic Web, Cloud Computing, Privacy

Keywords

Semantic cloud, Law-as-a-Service (LaaS), legal policies, privacy protection, stratified Datalog with negation, defeasible reasoning

1. INTRODUCTION

Cloud computing has become a generic term to describe an easy, flexible, and scalable delivery of resources and services over the Internet. This provides an emerging model in support of "everything-as-a-service"(XaaS); e.g., Software-as-a-Service (SaaS), and Database-as-a-Service (DaaS), etc. A new spectacular phenomenon of data sharing and service integration are possible in the cloud computing environment. Consequently, Integration-as-a-Service and Security-as-a-Service will also be available in the near future. This paves the way for technology innovation when applying semantic technologies to cloud computing. To leverage the mutual benefits arising from the combination of cloud computing and semantic web, three ways have been proposed to fulfill this vision: cloud computing for semantic technologies, semantic technologies offered as cloud services, and better clouds through semantic technologies [13].

However, current cloud infrastructures do not provide automated self-managed services. Therefore, a cloud provider's internal employees have to use manual service management, which requires intensive human intervention to explore and allocate the available virtual resources for cloud users [1]. Certainly this is not adequate when cloud resources are agile and deployed in the wide-scale of the Internet. We need self-managed SaaS for not only automated allocation of various cloud available resources, but also enforcing security and privacy policies without human intervention. *Law-as-a-Service (LaaS)* enhances self-managed SaaS on the automated security and privacy policy in the multi-tenant virtual data centers that provides law-aware semantic policies enforcement in the cloud [6].

In this paper, structure data are modeled as ontologies and used for data integration. This could leverage the abstract concept representation of ontologies, which do not exist in the relational database [8]. Furthermore, the stratified Datalog rules with exceptions handling capabilities extend ontologies to empower data protection and query services [5]. We have built a semantic data cloud for data integration and applied semantic web technology to represent semantic legal policies for data protection [25].

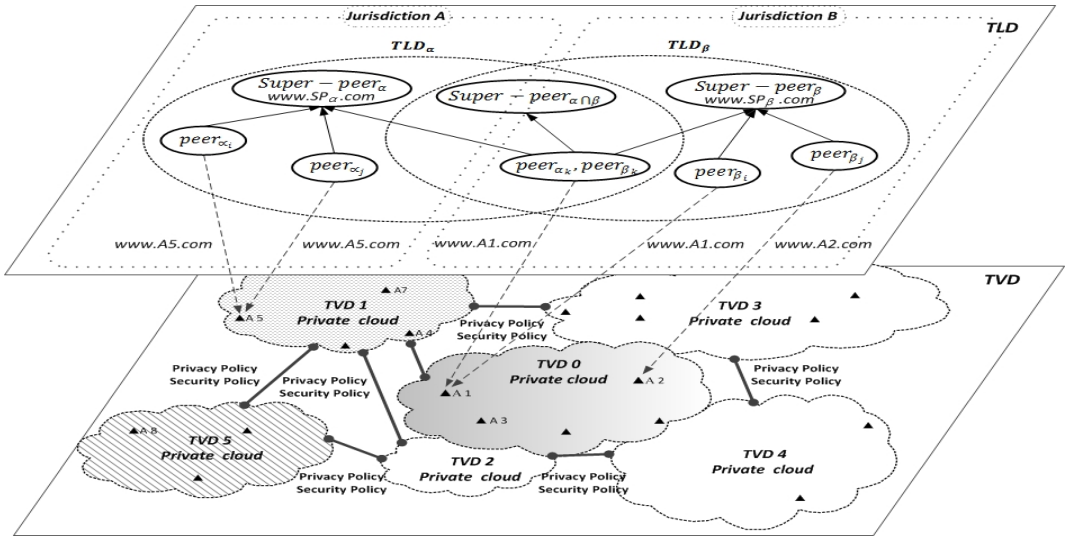

Figure 1: A conceptual layout of the law-aware semantic policy infrastructure in the cloud.

1.1 Semantic Policy Infrastructure

Inspired by a policy-aware Web concept [36], we propose a law-aware semantic policy infrastructure to provide LaaS for various cloud service providers (CSPs) and their potential customers. This paper extends our previous research work in [25]. In the previous study, we proposed a three-layer semantic policy infrastructure: Trusted Legal Domain (TLD), Trusted Virtual Domain (TVD), and Trusted Machine Domain (TMD). Here, we propose a super-peer domain in this policy architecture for modeling a specific Trusted Legal Domain (TLD)(see Figure 1).

Similar to a privacy appliance in [34], a super-peer is a unique law-aware guardian, that provides LaaS for its peers. The super-peer, also a trusted proxy of a super-peer domain, provides a query interface between its peers and a user. Peers own the real data sources, therefore the super-peer provides a data integration service for its peers. In addition, the super-peer specifies how law compliant legal policies are unified and enforced in a super-peer domain. The automated enforcement of unified legal policies protect data collected not only from its own peers but also from another super-peer domain's peers.

1.2 Logical Cage vs. Legal Cage Model

A TVD consists of a set of distributed virtual machines (VMs), storage for the VMs, and a communication medium interconnecting the VMs. A TVD provides a policy and containment boundary around the VMs [6]. In this paper, a semantic cloud of TLDs are established over the TVDs (see Figure 1). When a user asks for LaaS in the wide-scale semantic cloud, two objectives are achieved: first, a legal-unaware CSP, as a LaaS user, would use this service in legally helping the deployment of resources and services deployment. Second, after resources and services have been deployed and dispersed by CSPs, LaaS provides legalized query services for data users. Therefore, in the semantic law-aware policy infrastructure, automated data query from multiple data sources is available in a legally compliant way.

The LaaS has been implemented in the semantic policy infrastructure to verify this concept (see Figure 2). The TVD is a *logical cage* model. Corresponding to the semantic legal policies, uniform policies for security and privacy protection are created and executed in a TVD. We leverage the EU FP6 Open Trusted Computing (OpenTC) project results and build the semantic policy infrastructure on top of the TVD layer [6].

In this semantic policy infrastructure, the TLD is a *legal cage* model to circumscribe the virtual legal boundary of data usage. The TLD, determined by a specific law, regulates the range and level of data disclosure. TLD concepts are modeled as a taxonomy of laws in the policy ontology, where a type of law and an effective judicial domain are two factors to decide whether a data request is allowed to enter a specific TLD. Sometimes, a data request might enter an intersection of TLDs. In such case, the semantic legal policies are unified and executed to comply with the laws declared from these TLDs.

Legal policies are mapping from a data usage context to access control decisions, such as permit, deny, and error. A possible data usage context comprises a user's role along with his/her personal properties, resources metadata, access time, access location, purpose, and action, etc. A data usage context is created when a user asks for information from a TLD at the super-peer. Once a user's data usage context is satisfied with the domain policy of a TLD, the semantic legal policies of this TLD are identified and executed. The truth is legal policies are expressed as logical theories for information queries, and context are sets of ground facts (or instances) that fed into policies for outputs. Policy outputs (or query answers) are also encoded as logical formulas for authorization [5] (see Figure 2).

1.3 Research Issues and Contributions

Research issues. We identify several research issues for this study: (i) how to empower the semantic technologies for cloud computing to provide law-aware semantic cloud poli-

cies, (ii) how to use semantic legal policies to represent and interpret laws to ensure the legality of data sharing and access across jurisdictions, (iii) how to accomplish data protection while enforcing data integration, and (iv) how to unify semantic policies from TLDs to allow defeasible reasoning [33] for policy exceptions handling, and furthermore resolve possible conflicts among the unifiable legal policies.

Our contributions. Our main contributions are: (i) a law-aware semantic cloud policy infrastructure is established to verify the feasibility of the LaaS concepts, (ii) semantic legal policies are designed and enforced in a super-peer domain for data integration and protection, and (iii) constructing multiple super-peer domains to verify the possibility of data integration and protection across TLDs through unifying legal policies enforcement. In addition, the stratified Datalog with negation is used for a policy's exceptions handling.

Outline. This paper is organized as follows. In Section 2, we firstly introduce the background of a super-peer domain model. Then, three "queries as views" approaches for data integration are briefly explained. Furthermore, we address the stratified $Datalog^{\neg}$ for defeasible rules. In Section 3, related studies, such as LKIF and SemPIF, are presented. In Section 4, we present a law-aware semantic cloud and show how the principles of privacy protection laws are applied in the semantic cloud. In Section 5, we address the issues of the semantic model of TLD(s) for unambiguously enforcing semantic legal policies. Semantic legal policies are formally defined in Section 6. In Section 7, we unify two types of legal policies, derived from data protection and national security laws, to address the problem of exceptions handling for policy enforcement. In Section 8, we present how the legal policies are enforced, especially focus on defeasible reasoning for a policy's exceptions handling. In Section 8.3, we demonstrate how to apply our approach for a pandemic investigation scenario. Finally, in Section 9 we conclude this paper and point out the possible future work.

2. BACKGROUND

2.1 A Super-Peer Domain Model

The `PAYGO` and `Piazza` systems showed that a Peer Data Management System (PDMS) is the best way to achieve wide-scale data integration over the Internet [7] [22] [30]. This inspires us to put forth effort this concept for cloud computing. However, the `PAYGO` and `Piazza` systems only used a relational data model, but not used an ontology-based conceptual data model for wide-scale data integration. This hampers using conceptual-based information integration for describing real world entities.

A pure peer data integration architecture is hard to enact in the cloud environment because of the unstructured peer relationships nature for a large amount of peers. Thus, it will be a big challenge to provide unifying legal policies services in an unstructured peer data management system for effective data integration and protection.

We propose a super-peer data model to allow data integration and protection across jurisdictions. Within a super-peer domain, a super-peer specifies its legal semantic policies based on a type of law from a jurisdiction. Any peer registers at the super-peer, pledging to comply with the law declared as legal policies in the super-peer. We allow a peer registration at multiple super-peers, for each super-peer is in charge of a specific type of law on data sharing. This implies that when a peer is affiliated with multiple super-peers, the semantic policies declared in the super-peers are unified to enact data integration and protection for this peer. See Section 7 for more detailed information.

2.2 Queries as Views

In terms of data integration for multiple data sources, 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 [23]. The first approach, called global-as-view (GAV), requires that each concept in the global schema be expressed as a 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 approach, 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 [8] [28]. LAV descriptions handle the case in which the global schema contains details that are not present in every data source.

The third approach, called global-local-as-view (GLAV), is a source description that combines the expressive power of both GAV and LAV, and allows flexible schema definitions to be independent of the particular details of the data sources [17]. The data integration system uses these different source descriptions to reformulate a user query into a query over the source schemas. However, data integration is hampered by legitimate and widespread privacy concerns, so it is critical to develop a technique that enables the integration and sharing of data without losing a user's privacy [10] [31].

2.3 Stratified $Datalog^{\neg}$ for Defeasible Rules

Datalog is a database query language based on the logic programming paradigm: a set of ground facts, called the *Extensional Database (EDB)*, physically stored in a relational database, and a Datalog program P called the *Intensional Database (IDB)*. A Datalog program P is a mapping from EDB-facts database to IDB-facts. Stratified $Datalog^{\neg}$, which reduces data complexity and offers defeasible (or non-monotonic) reasoning, is an extension of pure Datalog with rule stratification and negation [9]. For a legal policy's exceptions handling, we use stratified $Datalog^{\neg}$ rules but not use defeasible DL-based ontologies. Because in general computational complexity is still very high by using defeasible OWL-DL reasoning [5]. Stratified Datalog with negation for multiple access control policies were also available in FAF [27].

3. RELATED WORK

This paper's research goal is quite different from the policy-aware Web, where they only consider achieving rule-based access control objectives [36]. We consider LaaS as an emergent technology, it not only accomplishes automated self-

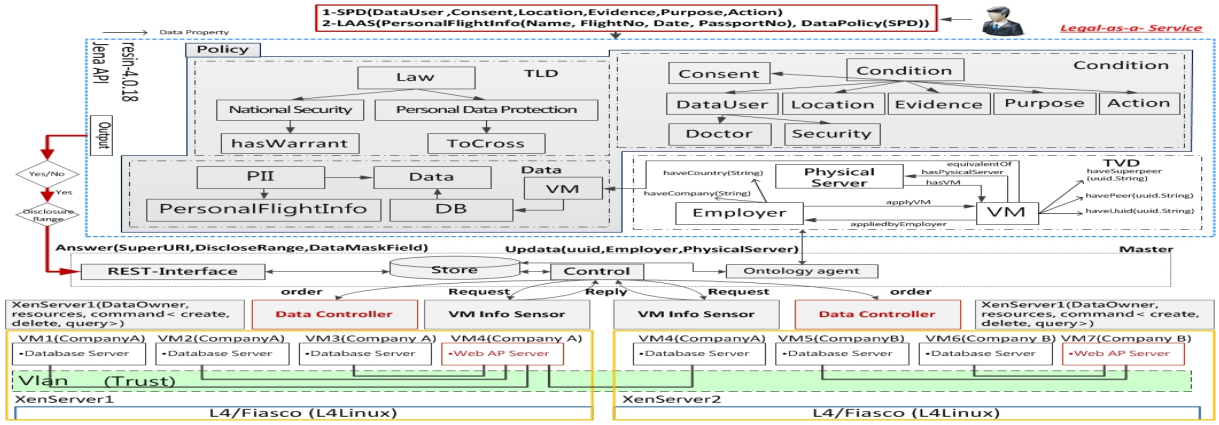

Figure 2: Semantic legal policies, expressed as logical theories for information queries, shown as a combination of OWL-DL ontologies and stratified Datalog rules with negation.

managed services for CSPs, but also ensures legal policies are compliant with the laws for users.

Semantic technologies for enterprise cloud management is another study, that uses semantic technologies to fulfil the challenges of intelligent information management, especially on the issues of data integration, intelligent information access, and analytics [20]. However, it did not consider enforcing legal policies to ensure the laws compliance while providing automated resources self-managed services.

The Legal Knowledge Interchange Format (LKIF) is a semantic web based language for representing legal knowledge to support modeling of legal domains [4]. Three types of knowledge are indispensable to legal reasoning in LKIF: terminological knowledge, legal rules, and normative statements. In our study, OWL-DL policy ontologies are used as terminological knowledge for legal norm representations, and $Datalog^{\neg}$ rules are used for a policy's exceptions handling through defeasible reasoning of the semantic legal policies.

In fact, LKIF rules extend SWRL with support for negation and defeasible reasoning. Meta-level information about rules is used for reasoning about rule priorities when rules conflict exists. A legal rule is defeasible if its conditions are true, unless some exception applies. However, a party who wants to apply this rule needs not show that the exception does not apply. The burden of producing evidence that the exception does apply is on the other party [19]. This will be demonstrated in Section 7.2, where a national security officer must provide legal evidence to satisfy a closed policy exception constraint before requesting a personal data without a data owner's previous consent.

In SemPIF [24], a meta-policy is a policy about policies that provides a set of rules for realizing services needed for the management of policies. Moreover, a meta-policy consists of a set of rules for setting up the priority of polices to coordinate, enforce, and even negotiate policies. However, meta-policy is unlike Datalog, it is only used for policy conflict resolutions but not for a defeasible inheritance within ontologies and rules.

In [2], privacy policies were expressed as a first-order logic. The idea of privacy expectation can be stated using context information norms. A information flow satisfies privacy expectations if any one positive norm and all negative norms applicable to the context of the transmission are satisfied. Both positive norms and negative norms may also contain exceptions. A positive norm's exceptions simply refine the norm's specific conditions. On the other hand, a negative norm's exceptions provide a choice for a norm satisfaction [11].

4. LAW-AWARE SEMANTIC CLOUD

A policy-aware infrastructure was defined as an integral part of the Semantic Web in [36]. It gives users greater transparency in their online interactions and helps both people and machines to 'play by the rules' relevant to social interactions. Thus, policy awareness provides users with readily accessible and understandable views of the policies associated with resources and makes compliance with stated rules easy. The transparency, compliance, and accountability requirements are satisfied in this policy-aware infrastructure.

We intend to achieve similar objectives as [36]. However, we focus more on the issue of enforcing law-aware policies in the semantic cloud infrastructure. The TLD layer applies the semantic technologies over the trusted cloud virtual infrastructure (or platform), such as OpenTC cloud, to fulfill two visions:

1. It offers LaaS for CSPs while integrating semantic data modeled as ontologies from multiple data sources. The law-aware semantic cloud services help CSPs spot and track infraction when they plan to deploy their resources and services. LaaS also provides CSPs with transparent semantic policies that are compliant with the current up-to-date laws in the cloud.

2. The ontologies and stratified Datalog rules with negation are used for representing legal policies to enable query services for real cloud end-users. The legal policies are automatically enforced by the systems because metadata extracted from the semantic data cloud are used in deciding whether or not the integrated data

satisfy with the legal policy's preconditions. If the data usage context satisfies the preconditions, data are *disclosed*. Otherwise, they are hidden (or $\neg$*disclosed*). Therefore, all of the data integration and query services are ensured to comply with the current laws.

In addition, we also offer another service requirement: flexibility. This allows CSPs to configure their executable legal policies and re-engineer their cloud management services whenever the laws that regulate the cloud computing environment are updated.

4.1 The Principles of Data Protection Laws

When we implement the cloud computing environment, major challenges encountered include: security, privacy and conformity to laws. Law inevitably tends to lag behind technology, yet dramatic technological changes can alter the scope of laws overnight. In fact, the main legal issues related to cloud computing are data protection and security, which are by far the biggest concerns for CSPs and their customers.

As for data protection legal issues, processing of personal data in Europe is mainly regulated by the Data Protection Directive 95/46/EC, which is currently under revision. In a legally uncertain situation, in order to handle the legal policies in compliance with the emerging data protection laws, we propose flexible relationship mapping between TLD and TVD layers. This allows us to dynamically enable/disable the security and privacy policies on the TVD layer using our self-managed LaaS (see Section 1).

Whenever the laws used for regulating cloud computing are updated and expecting to unify with the other laws from different jurisdictions, the semantic legal policies for a TLD are modeled as a combination of revised ontologies and rules, then re-mapped to the updated security and data policies in TVDs. Therefore, we ensure our law-aware semantic cloud policies are always in compliance with the most up-to-date laws for cloud operations.

The principles of using privacy protection laws in the cloud depend on three criteria:[1] (1) the registration principle: the location of service provider registration, which enables data collection services, (2) the nationality principle: the nationality of the data owner whose data are being used, and (3) the territoriality principle: the data center location where actual data processing happens.

Currently, people do not have consensus on which principles are used for enforcing the privacy protection laws in the cloud, especially across jurisdictions. In this paper, we offer LaaS for CSPs before deploying their cloud resources, and allow them the flexibility to choose any principles of the privacy laws with which they prefer to comply. We also ensure that all of the subsequent queries on the resources and services deployment cloud are in compliance with the principles of selective laws. We manually unify legal policies to avoid any possible conflicts of data disclosure across jurisdictions.

[1] Fleischer, P., Which privacy laws should apply on the global Internet?, `http://peterfleischer.blogspot.com`

5. SEMANTIC SUPER-PEER DATA CLOUD

In a pure peer data cloud, each peer exports data in terms of its own schema. Information integration is achieved by establishing mappings among various peer schemas [21]. This unstructured data integration is too complex for heterogeneous peer schemas when the number of peers are large. In the super-peer network architecture, we group a set of peers into a super-peer domain and organize them into a two-level architecture as in [3]. In the lower level, called peers, and in the upper level, called the super-peer.

More precisely, a peer transforms from the relational data sources in a TVD into a local ontology in a TLD. The super-peer is a guardian of data integration system, which integrates all of its local peers' ontologies into a global ontology through ontology mapping, alignment, and merging [15] [26]. Therefore, a traditional data integration system is viewed as a special case of a super-peer data model.

The emergent semantics establishment in a super-peer data cloud allows flexibly integrating data from another super-peer domain by using the semantic mapping technique. Information is requested on demand from the intersection of TLDs. This wide-scale data integration problem faces the challenge of effectiveness data sharing without causing semantic ambiguity of ontology mappings among super-peers.

We propose a wide-scale data integration and protection architecture in a super-peer data cloud. Any peer from a super-peer domain can contribute new data, schema, or even mappings through its super-peer to other super-peer domains for data integration. We define a super-peer data cloud system as a set of super-peer domains $\Pi = \{\pi_1, ..., \pi_n\}$, where each super-peer domain π_i corresponds to a TLD. It is an autonomous information site that exports its data in terms of the super-peer sp_i's schema to another super-peer domain.

Each super-peer domain is essentially a mediator-based data integration system, where an $agent_\alpha$ at the $Super-peer_\alpha$ performs semantic local mappings to manage a set of its local peers endowed with shareable relational data sources. Semantic global mappings are also possible from the current $Super-peer_\alpha$ to interlink with another $Super-peer_\beta$, where $agent_\beta$ is in charge of data integration and protection services for $Super-peer_\beta$. Through semantic legal privacy policies enforcement, authorized view-based queries are posed to a super-peer that provides data integration services.

5.1 Semantics of a TLD

In a TLD_α's super-peer domain π_α, actual data is stored in a set of local relational data sources $DS_\alpha = \{ds_1, ds_2, ..., ds_m\}$. Using GAV local mappings, we associate a set of local peer $P_\alpha = \{peer_1, peer_2, ..., peer_n\}$ in π_α with each individual ontology schema to the views of the related relational data sources, i.e., SQL queries. Furthermore, through LAV semantic mappings, a set of peers' local ontology schemas are also mapped and aligned into the super-peer sp_α's global view.

A super-peer domain $\pi_\alpha \in \Pi$, corresponding to a TLD_α, can be defined as a tuple $(P_\alpha, SPD_\alpha, GS_\alpha, LS_{peer_i}, M_\alpha, DS_\alpha)$:

- A super-peer sp_α is the only node in a super-peer domain $\pi_\alpha \in SPD_\alpha$, which allows an $agent_\alpha$ to enforce the semantic legal policies. This enforcement action empowers $agent_\alpha$ in the super-peer sp_α to facilitate information collection through a conjunctive query $CQ_{\pi_\alpha}(sp_\alpha)$ posed to the global schema GS_α in the super-peer sp_α of π_α. A $CQ_{\pi_\alpha}(sp_\alpha)$ can be defined as a subset of the Datalog program, i.e., a CQ containment problem, for querying the relational database [18].
- Through the local LAV mapping assertions, a global schema GS_α provides an integrated view for a set of peers from P_α in a π_α. We proposed that every LAV assertion has the form $V_{LS_{peer_i}} \rightsquigarrow CQ_{\pi_\alpha}(sp_\alpha)$, where $V_{LS_{peer_i}}$ provides the views of the $CQ_{\pi_\alpha}(sp_\alpha)$ over the global schema GS_α at the super-peer sp_α for $peer_i$.
- A set of peers from P_α are mediators. A peer $p_i \in \pi_\alpha$ maps its local ontology schema, LS_{peer_i}, to a set of relational data sources, ds_i, from DS_α in π_α. Therefore, this query uses the unfolding GAV mapping assertions. $V_{LS_{peer_i}} \rightsquigarrow CQ_{\pi_\alpha}(DS_\alpha)$, where $V_{LS_{peer_i}}$ is the vocabulary of an ontology local schema of a $peer_i$ that maps to the SQL of $CQ_{\pi_\alpha}(DS_\alpha)$ over a set of data sources, ds_i, from DS_α.
- A set of local mapping assertions, M_α, created from a mapping language, ML, are used to semantically link between a super-peer sp_α and a set of peers from P_α in a π_α. The semantics of a set of global mapping assertions created from a Datalog rule language among super-peers will be addressed in Section 5.2.
- A set of local data sources, ds_i, from DS_α, are relational structure data that store the materialized instances.

5.2 Semantics of Multiple TLDs

When LaaS supports cloud resource deployment and queries across TLDs, laws declared in each TLD should be unified in order to comply with all of them. A super-peer domain π_α for TLD_α is related to another super-peer domain π_β for TLD_β by means of a set of super-peer GLAV semantic mapping assertions. GLAV is the most powerful source description that combines GAV and LAV schema mapping descriptions, which offers a flexible schema mapping assertion [17]. A super-peer semantic schema mapping assertion between TLD_α and TLD_β is shown as follows:

$$CQ_{\pi_\beta}(sp_\beta) \rightsquigarrow CQ_{\pi_\alpha}(sp_\alpha)$$

where $CQ_{\pi_\beta}(sp_\beta)$ is a conjunctive query over the super-peer sp_β in a super domain $\pi_\beta \in \Pi$, and $CQ_{\pi_\alpha}(sp_\alpha)$ is a conjunctive query over the super-peer sp_α in a super domain $\pi_\alpha \in \Pi$. A $CQ_{\pi_\beta}(sp_\beta)$ is defined as an authorized legal view of a super-peer domain π_β whenever the super-peer sp_β intends to export its data through unifying its legal policies with another super-peer domain π_α. The global schema GS_β of sp_β is mapped to another sp_α's global schema GS_α through the super-peers' GLAV semantic mapping assertions. However, *law-unaware* super-peers are different from *law-aware* super-peers on modeling the semantics of unifying legal policies when they enable data integration and protection services in multiple TLDs. For more detailed information shown as follows:

When data queries across super-peers impose on the legal-unaware super-peer, prearrangement mapping assertions are not required for the GLAV schema mappings. A Datalog rule, r_i, is used for mapping assertions of GLAV:

$$H \longleftarrow B_1 \wedge B_2 \wedge, \cdots, \wedge B_n,$$

where H, the query results (or views) are from the source of sp_α's global ontology schema, and the rule antecedent B_i, is defined as a pattern matching specifications, i.e., query conditions that are class and property vocabularies. These are from the target sp_β's global ontology schema.

In the simplest one-to-one mapping assertion, the Datalog rule corresponds to $\pi_\beta \in \Pi$, containing a mapping from the super-peer sp_β's shareable ontology global schema vocabulary, R_β, to another randomly select super-peer sp_α's shareable ontology global schema vocabulary, R_α. One edge from the super-peer sp_β's vocabularies R_β to the super-peer sp_α's vocabularies R_α exists if there is a super-peer mapping assertion available in Π whose tail mentions R_β and whose head mentions R_α in a Datalog rule, i.e., $R_\alpha \longleftarrow R_\beta$. On the other hand, above Datalog rule used for mapping assertions of GLAV is enacted in the many-to-many mapping assertions.

Conversely, when the queries go through the intersection of TLDs across *law-aware* super-peers, we specifically unify the pre-arranged legal policies to discover the mapping assertions from the vocabulary of sp_β's global ontology schema to the vocabulary of sp_α's global ontology schema. Furthermore, potential policy conflicts between these unifiable legal policies should be resolved by Datalog rules with rule priority setting. Or a legal policy's exceptions are handled by defeasible reasoning from the stratified $Datalog^\neg$ rules shown in Section 8.3.2.

6. SEMANTIC LEGAL POLICIES

Lessig presented the idea that regulation through laws on the Internet is often inefficient. Therefore, forcing users to behave in a legally compliant way through embedding legal concepts directly into the computer code is another feasible alternative [29]. Based on our previous studies [25] [26], we propose the semantic legal policies as a combination of ontologies and rules, where ontologies represent the concepts of laws that are executable in the semantic cloud and rules enforce data query and protection services in a TLD or across TLDs.

6.1 Legal Policy Representation

A formal *legal policy* is a declarative expression executed in a computer system for a human legal norm without causing semantic ambiguity. A *legal policy* is created from a *policy language*, and a *legal policy language* is shown as a combination of ontology language and rule language. A *legal policy* is composed of ontologies and rules, where ontologies

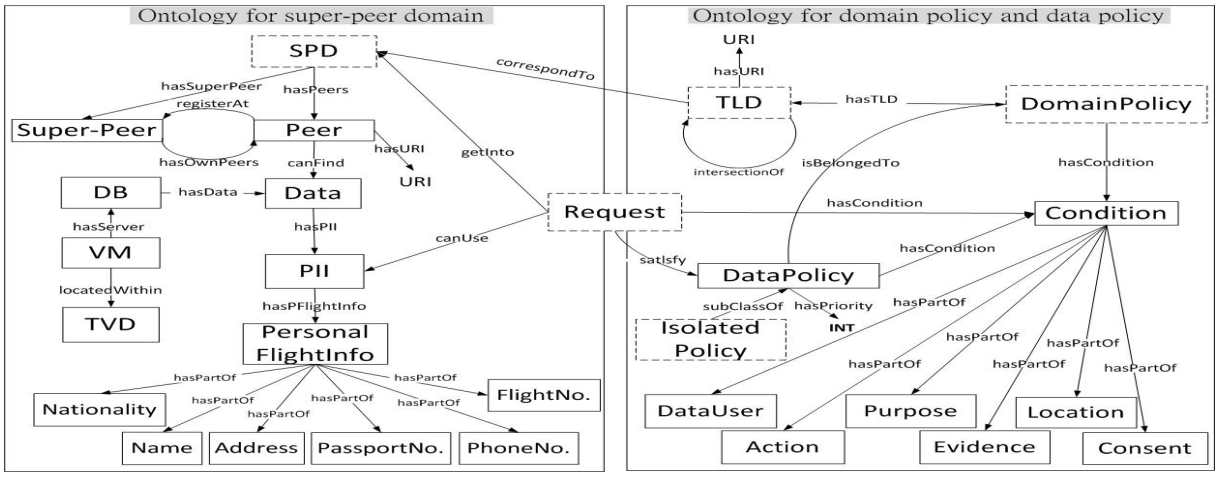

Figure 3: Semantics of a super-peer data cloud is described as policy ontology, which includes two modular concepts: (1)super-peer domain, (2)domain policy and data policy. Super-peer domain includes a super-peer and various peers. A domain policy first decides whether a data request is permitted to enter a TLD. Then, a data policy is used to query data from the super-peer.

are created from an ontology language and rules are created from a rule language.

Furthermore, a *legal protection policy* is a *legal policy* that aims at representing and enforcing the privacy protection principles of resources in the semantic cloud, where the structure of resources is modeled as ontologies and the protection of resources is shown as rules. The privacy policy model used for access control in enterprises has been extensively investigated in [35], where only Logic Program (LP)-based Datalog was used for designing the privacy policy.

In this study, a *privacy protection policy* is a combination of ontologies and rules, where Description Logic (DL)-based ontologies provide data integration, while Logic Program (LP)-based rules provide data query and protection services after data integration. A combination of ontologies and rules in the semantic web can refer to [14]. SWRL or OWL2 are semantic web languages, which provide a homogeneous combination of DL-based ontologies and LP-based rules to represent legal protection policies. A *global policy schema* allows data integration through unifying of regular *policies* from a variety of structure data sources, where *global policy schema* includes integrated ontologies and rules.

We proposed a semantic privacy-preserving model for data sharing and integration using SWRL in [26]. The soundness and completeness criteria of this model were specified and proved. We extend our previous study results in this paper. The semantic privacy-preserving model is applied to a super-peer domain, where only the law-aware super-peer could enable data integration and protection services. In addition, we provide laws complying data integration services through *unifying legal policies* between super-peers. When the stratified Datalog with negation, $Datalog^{\neg}$, rules are used for defeasible rules reasoning, the research challenging is how to integrate the two families of logics, DL and LP, for legal policy enforcement under a non-monotonic semantics. Expressiveness is not the only issue, as hybrid integration usually have high computational complexity. The appropriateness of the above hybrid integration to the design of policy languages for privacy protection policies has not yet been completely studied [5].

Current data protection laws in the semantic cloud are not up-to-date on handling data sharing and protection across jurisdictions. We address the research issues, not only for law refinement, but also for technology re-engineering when the concepts embarked in the laws for regulating the cloud are updated. The ultimate objective of this study is to empower the flexible and agile use of cloud resources without fear of violating data protection and other related laws.

6.2 Legal Policy Compliance

Legal policy compliance addresses the issues for data access and service execution in the semantic data cloud. The semantic legal policies enforcement should satisfy up-to-date laws in a jurisdiction. However, the resources, data, and services in the cloud are usually dispersed over the Internet. So anyone can dynamically request anything from anywhere at anytime. In this case, we might have to unify the laws across jurisdictions to regulate a data request. This raises the regulation compliance issue for how to ensure the formal legal policies are enforced, which satisfies the data usage context derived from the related laws.

The legal policy compliance of each data request is based on the *data usage context* created for each user. It is a precondition when applying the laws for a query in a TLD. In the policy ontology (see Figure 3), whenever a user's data usage context is within a TLD's *domain-policy*'s *Condition*, e.g., the concept of a data usage context is subsumed by a domain policy's context, this data request enters this specific TLD. We are in compliance with the laws of a *domain-policy* because of a data usage context subsumption. After a domain-policy is chosen, an applicable *data-policy* belonged to this TLD is initiated to enable real data access. However, this data usage is only used for a single legal domain TLD.

When a data usage is across jurisdictions, such as the intersection of data protection and national security TLDs, we need an iterative legal policy enforcement process to achieve selective revelation of anonymous personal identifiable information (PII). It has been unusually challenging to build a legal framework for protecting individual privacy in the struggle against terrorists since the 9-11 terrorist attacked on U.S. [32] [34]. The widespread of cloud computing services certainly will exacerbate this challenge. We attempt to address this research issue and provide one of the possible solutions based on our semantic legal policy enforcement.

7. UNIFYING LEGAL POLICIES

We propose a semantic legal policy framework to serve flexible policy deployment, integration, and enforcement. In this policy framework, legal policies representing privacy protection law α and national security law β are unified at a $Super-peer_{\alpha\cap\beta}$ of a $TLD_{\alpha\cap\beta}$, where a super-peer domain of $TLD_{\alpha\cap\beta}$ is in the intersection of TLD_α and TLD_β jurisdiction (see Figure 4).

In terms of the national security law, the Electronic Communication Privacy Act (ECPA) was enacted in 1986 and did not foresee the proliferation of cloud computing [12]. Therefore we need to reform the ECPA to balance the personal privacy rights and government law enforcement needs in the cloud computing environment.

7.1 A Peer Registers at a TLD

Before a peer, p_i, offers its data for integration in a super-peer domain, SPD_α, p_i should register at a super-peer sp_α. The sp_α is in charge of the SPD_α shown as TLD_α for data integration and privacy protection services. The sp_α specifies a data usage context, which is in compliance with the privacy protection laws for all of the peers in the SPD_α. The registration action implies that p_i pledges to obey the privacy protection law by applying specifications for the data usage context declared in the sp_α.

Based on this data usage context specification, p_i uses LAV source description to export its data to the sp_α for data integration. In case a peer p_i also registers at another super-peer domain, SPD_β, shown as TLD_β, and exports its data for national security policy enforcement purposes. This indicates that the laws from sp_α and sp_β, which are shown respectively as privacy projection and national security legal policies, are unified and enforced after collecting data from p_i.

Sometimes, the open cloud is constructed as a virtualization layered architecture for multi-tenant services. A peer is a virtual node within a super-peer domain, and it corresponds to a database installed in virtual machines (VMs). We might face a situation, where the database is in compliance with a data protection law α from one jurisdiction but the data center providing VMs for hosting the database is in compliance with a national security law β from another jurisdiction. This multi-tenant cloud service layout is different from the Gmail and Facebook situations, where the cloud management services of data center and database belong to a single legal authority.

One possible solution to resolve this legal discrepancy is to enact service level agreement (SLA) between the owners of database and data center before the database is installed in the data center's VMs. The SLA provides necessary information for the database owner to ensure that (s)he is aware of this legal domain discrepant situation. Another possible solution for preserving privacy in data outsourcing is to enforce privacy over data collections combining *data fragmentation* with *encryption* to avoid any illegal data request from *curious but honest* cloud providers [16]. Unless the national security law enforcement officers are complied with the SLA and respective national security laws, any data disclosure request without a data owner's previous consent will be rejected. More detailed information see Section 8.3.

Based on the above proposition, we propose a solution through unifying legal policies submitted from different judicial authorities. On the one hand, when an end-user asks for data from the $TLD_{\alpha\cap\beta}$, two types of legal policies, privacy protection and national security, from different judicial domains are unified to legally restrict data access at p_i. On the other hand, when an end-user asks for data from sp_α or sp_β separately, we do not unify legal policies in this situation, so only one type of law is applicable for this data request.

7.2 Query at the Intersection of TLDs

In Figure 4, an agent α in TLD_α performs privacy protection law and an agent β in TLD_β performs national security law. When a data usage context from agent β firstly satisfies the conditions of national security law enforcement, such as a data user's role as a national security officer, a data owner's consent is absent, and the purpose of data disclosure is for national security, then we enter the $TLD_{\alpha\cap\beta}$ legal domain. We model the national security law enforcement as the privacy policy's exceptions. Whenever a national security officer queries data at the $Super-peer_{\alpha\cap\beta}$, the nationality principle shown in Section 4.1 allows another jurisdiction's privacy protection law to bend. However, only anonymous PII will be disclosed. This is because we still have to somehow ensure the privacy protection law α is not violated. This approach balances personal privacy rights and national security needs in the cloud.

We manually unify two types of legal policies, which are translated from the privacy protection law and the national security law to demonstrate how data is collected from peers who have registered at the $Super-peer_{\alpha\cap\beta}$. Two types of queries are available: subject-based queries and pattern-based queries. At the $Super-peer_{\alpha\cap\beta}$, we only provide pattern-based queries. This contrasts the queries provided at the $Super-peer_\alpha$ and $Super-peer_\beta$, where we provide both. Similar to the privacy appliance in [34], a trusted agent at the $Super-peer_{\alpha\cap\beta}$ is a guardian, who follows the laws and provides privacy protection and national security legal services while disclosing data from its registered peers within the $TLD_{\alpha\cap\beta}$.

The guardian in the $Super-peer_{\alpha\cap\beta}$ only grants anonymization pattern-based queries so that initially a PII cannot be fully discovered. This process proceeds incrementally by requiring the $Super-peer_{\alpha\cap\beta}$ to release a subset of a PII. So the filtering mechanism embedded in the guardian only reveals some of a PII's attributes. Through legal reasoning of unifiable legal policies with additional information linkage,

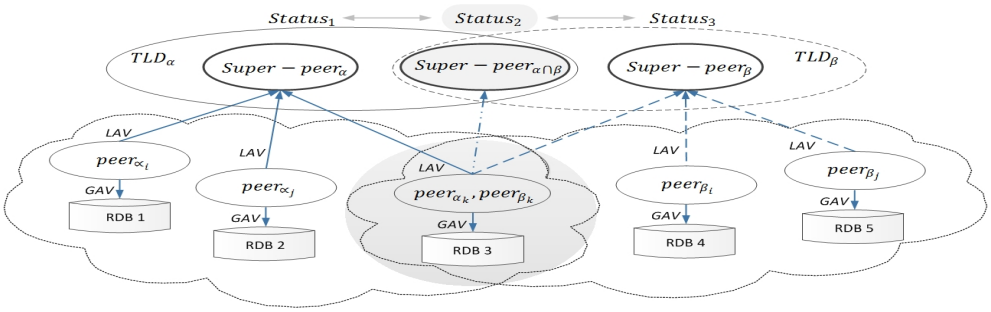

Figure 4: A data request uses its own data usage context to decide which super-peer to contact. Each super-peer binds a type of law for data access in its super-peer domain. For example, the $Super-peer_\alpha$ uses the data protection law in TLD_α, and the $Super-peer_\beta$ uses the national security law in TLD_β. Finally, a query abides by the unifying laws at the $Super-peer_{\alpha\cap\beta}$ in $TLD_{\alpha\cap\beta}$.

a few individuals' PII with plausible cause are finally permitted to be fully disclosed. However, we should re-initiate a new query at the $Super-peer_\beta$ so this situation only happens in the TLD_β.

In summary, we manually unify privacy protection legal policies with national security legal policies not only to ensure privacy, but also to encourage sharing data for national security without fear of a privacy rights violation. We use pattern-based data queries so that the anonymous disclosure principles of PII can be applied at the $TLD_{\alpha\cap\beta}$ to minimize the false positives.

8. LEGAL POLICY ENFORCEMENT

8.1 Enforcement in a TLD

We proposed a semantic privacy-preserving model, which provides authorized view-based queries over a relational data sources from a widespread of autonomous multiple servers [26]. We also used the semantic privacy protection policies to empower the data integration and access control at the three-layer virtual platform.

The regular policy combination was shown as ontology mapping, merging, and rule integration. The ontology mapping and merging algorithm creates a top-layer global ontology schema at the virtual platform by integrating middle-layer multiple local ontology schemas for data integration. Furthermore, the perfect Datalog rule enhances the ontologies to serve data query and protection on the virtual platform. The legal policy enforcement for data integration and protection was established in a super-peer domain for a TLD_α [25].

We further enhance this legal policy enforcement to unify legal policies in an intersection of TLDs. The policy ontology demonstrates how we configure the TLD into a super-peer domain and use the super-peer to offer data integration and protection services through unifiable legal policies reasoning (see Figure 3).

8.2 Defeasible reasoning for policy exceptions

In formalizing access control policies, we might deal with the situation where a give request is neither explicitly allowed nor explicitly denied. Then, a default decision has to be taken, such as in the *open* and *closed* policies, where authorization are respectively granted or denied by default. The layers induced by Datalog stratification may be regarded as the steps of a methodology for constructing privacy protection closed policies in a principled way, starting with explicit authorizations, then adding derived authorizations through inheritance along hierarchies of subjects, objects, purposes, and rules. Finally, it fills in policy gaps by introducing exceptions. This approach is clearly a defeasible inheritance.

We apply stratified $Datalog^\neg$ to deal with defeasible inheritance when legal policies are unified in the $Super-peer_{\alpha\cap\beta}$. Because the complexity of defeasible DL reasoning in general is still very high, and the major DL reasoning engines do not support defeasible reasoning [5].

8.3 A pandemic investigation scenario

Example 1. The α Inc. is an international airline company with headquarters located in Singapore. The α airline Inc. applies the *closed policies* of privacy protection laws, based on the registration and nationality principles described in Section 4.1. The first policy exception, Ab1, is no personal data disclosed unless a data owner's former consent was obtained.

Whenever a data disclosure request comes from the data owners' national security officer to enforce the national security, as long as it follows legal procedure with supporting evidence, this request will be granted without data owners' consent. So the second policy exception, Ab2, occurs in this situation. The α airline Inc. pledges to follow Singapore data protection laws but allows the second exception for data disclosure when any national security officer requests his/her own citizen flight information. However, the third exception, Ab3, is any national security cannot request an alien citizenship's data unless a data owner's former consent is positive.

A super-peer domain SPD_α is created for the α airline in compliance with Singapore data protection laws in TLD_α, where queries request data from its dispersed data centers around the world.

Another super-peer domain SPD_β is created for national security officers β of Taiwan CDC to enforce the *open policies* of national security laws for a pandemic investigation based

on the territoriality and nationality principles. Data disclosure exceptions are also possible for a national security law enforcement when the citizen nationalities are not Taiwan. Then these data cannot be disclosed unless a data owner's former consent was obtained. During the recent H1N1 pandemic period, a national security officer in Taiwan tried to trace H1N1 contagious persons, who possibly took inbound flights from Singapore to Taiwan within the last fourteen days.

How do we unify the legal polices enforced in two different jurisdictions and avoid possible legal policy conflicts through exceptions handling? Furthermore, what level and range of data are permitted to be disclosed when a query is initiated at a different super-peer, e.g., $Super-peer_{\alpha}$, $Super-peer_{\beta}$, or $Super-peer_{\alpha\cap\beta}$?

8.3.1 legal reasoning in SPD_{α}

Under normal conditions, we must go through the super-peer sp_{α} for subject or pattern-based queries in the SPD_{α} of α airline to enforce the legal policies of data protection laws and to ensure its legitimacy. For a legal policy enforcement in SPD_{α} for TLD_{α} using subject-based query shown as follows:

According to the policy ontology (see Figure 3), when a data request ?r with its data usage context ?c satisfy a DomainPolicy(?dmp)'s data usage context ?dmc. A user is allowed to enter the TLD(?tld) enforcing the listing of flight passenger (see rules (1) and (2)):

- A partial ontology for a domain policy:

  hasTLD.DomainPolicy(dmp),
  hasTLD^{-}.TLD(tld).

  Above two expressions indicate that property hasTLD has *domain* of class DomainPolicy and *range* of class TLD. Similarly, the hasCondition, hasPartOf, and other properties are shown as follows:

  hasCondition.DomainPolicy(dmp),
  hasCondition^{-}.Condition(dmc).

  hasPartOf.Condition(dmc),
  hasPartOf^{-}.Purpose(checkIn),
  hasPartOf^{-}.DataUser(airlineStaff),
  hasPartOf^{-}.Action(read).
  hasPartOf^{-}.Location(TW),
  hasPartOf^{-}.Consent($\top$).

  $= 1\ \text{hasSuperPeer}^{-}.\text{Super}-\text{Peer}(\text{sp})$,
  $\exists$hasPeers.Peer(p),
  $\forall$registerAt.Peer(p),
  $\exists\text{registerAt}^{-}.\text{Super}-\text{Peer}(\text{sp})$.

  This part of the ontology indicates that each super-peer domain has only one super-peer, and at least one peer. In addition, all of the peers must register at least one super-peer.

- Rules for a domain policy enforcement:
  The rule (1) provides a link between abstract trusted-legal domain (TLD) concept and concrete super-peer domain (SPD). In the rule (2), we decide which SPD a request is taken that depends on whether the request usage context is subsumed by a domain policy's context information.

  $\text{DomainPolicy}(?\text{dmp}) \wedge \text{hasTLD}(?\text{dmp}, ?\text{tld})$
  $\wedge\ \text{correspondTo}(?\text{tld}, ?\text{spd}) \wedge \text{SPD}(?\text{spd})$
  $\longrightarrow \text{domainPolicyForSPD}(?\text{dmp}, ?\text{spd}) \longleftarrow$ (1)

  $\text{Request}(?\text{r}) \wedge \text{hasCondition}(?\text{r}, ?\text{c}) \wedge \text{Condition}(?\text{c})$
  $\wedge\ \text{DomainPolicy}(?\text{dmp}) \wedge \text{hasCondition}(?\text{dmp}, ?\text{dmc})$
  $\wedge\ \text{Condition}(?\text{dmc}) \wedge \text{isSubsumed}(?\text{c}, ?\text{dmc})$
  $\wedge\ \text{domainPolicyForSPD}(?\text{dmp}, ?\text{spd})$
  $\longrightarrow \text{getInTo}(?\text{r}, ?\text{spd}) \longleftarrow$ (2)

  An ontology and a rule, for a data policy ?dap in the $\text{SPD}_{\alpha}(?\text{spd})$ for $\text{TLD}_{\alpha}(?\text{tld})$, allow a request ?r using the PII ?pii of personal information as follows (see rules (3)-(6)):

- A partial ontology for a data policy:

  isBelongedTo.DataPolicy(dap),
  isBelongedTo^{-}.DomainPolicy(dmp).
  hasPII.Data(da),
  hasPII^{-}.PII,
  hasPFlightInfo.PII(pii),
  $\text{hasPFlightInfo}^{-}$.PersonalFlightInfo(fInfo).

  hasPartOf.PersonalFlightInfo(finfo),
  hasPartOf^{-}.Name(name),
  hasPartOf^{-}.PassportNo.(pano),
  hasPartOf^{-}.Nationality(citizenship),
  hasPartOf^{-}.FlightNo.(fno),
  hasPartOf^{-}.Date(date).
  hasPartOf^{-}.Address(addr).
  hasPartOf^{-}.PhoneNo.(pono).

- Rules for a data policy enforcement:

  $\text{SPD}(?\text{spd}) \wedge \text{hasSuperPeer}(?\text{spd}, ?\text{sp})$
  $\wedge\ \text{Super}-\text{Peer}(?\text{sp}) \wedge \text{hasPeers}(?\text{spd}, ?\text{p})$
  $\wedge\ \text{Peer}(?\text{p}) \wedge \text{registerAt}(?\text{p}, ?\text{sp})$
  $\longrightarrow \text{hasOwnPeers}(?\text{sp}, ?\text{p}) \longleftarrow$ (3)

  $\text{Super}-\text{Peer}(?\text{sp}) \wedge \text{hasOwnPeers}(?\text{sp}, ?\text{p})$
  $\wedge\ \text{Peer}(?\text{p}) \wedge \text{canFind}(?\text{p}, ?\text{da}) \wedge \text{Data}(?\text{da})$
  $\wedge\ \text{hasPII}(?\text{da}, ?\text{pii}) \wedge \text{PII}(?\text{pii})$
  $\longrightarrow \text{hasDisclosedFor}(?\text{sp}, ?\text{pii}) \longleftarrow$ (4)

  $\text{DataPolicy}(?\text{dap}) \wedge \text{isBelongedTo}(?\text{dap}, ?\text{dmp})$
  $\wedge\ \text{DomainPolicy}(?\text{dmp})$
  $\wedge\ \text{domainPolicyForSPD}(?\text{dmp}, ?\text{spd})$
  $\longrightarrow \text{dataPolicyForSPD}(?\text{dap}, ?\text{spd}) \longleftarrow$ (5)

  $\text{Request}(?\text{r}) \wedge \text{getInTo}(?\text{r}, ?\text{spd})$
  $\wedge\ \text{satisfy}(?\text{r}, ?\text{dap}) \wedge \text{DataPolicy}(?\text{dpa})$
  $\wedge\ \text{dataPolicyForSPD}(?\text{dap}, ?\text{spd})$
  $\wedge\ \text{SPD}(?\text{spd}) \wedge \text{hasSuperPeer}(?\text{spd}, ?\text{sp})$
  $\wedge\ \text{hasDisclosedFor}(?\text{sp}, ?\text{pii})$
  $\longrightarrow \text{canUse}(?\text{r}, ?\text{pii}) \longleftarrow$ (6)

8.3.2 *Defeasible reasoning in* $SPD_{\alpha\cap\beta}$

Once a Taiwan national security officer enters a super-peer domain of $SPD_{\alpha\cap\beta}$, it must simultaneously comply with the Singapore data protection laws α, as closed policies, and the Taiwan national security laws β, as open policies. Here, we apply stratified $Datalog^{\neg}$ in the rule (2) for policy exceptions handling to comply both of the above laws. In closed-world-assumption (CWA) semantics, absent of consent, e.g. $\bot$, is weak negation.

Three exceptions (strata) are possible. In stratum one, by default, we do not disclose a personal dataset `A`, unless a data owner's explicit former consent was true, e.g. $\top$.

`Ab1` $\rightarrow$ `hasPartOf.Condition(Ab1)`,

`hasPartOf.Condition(Ab1)`,

$$Ab1 = \begin{cases} hasPartOf^{-}.Purpose(\neg nationalSecurity) \\ hasPartOf^{-}.DataUser(\neg securityOfficer) \\ hasPartOf^{-}.Consent(\top) \end{cases}$$

In stratum two, we permit the Taiwan national security officers to enforce a dataset `C` disclosure even lack of a data owner's explicit former consent, e.g., $\bot$.

`Ab2` $\rightarrow$ `hasPartOf.Condition(Ab2)`,

`hasPartOf.Condition(Ab2)`,

$$Ab2 = \begin{cases} hasPartOf^{-}.Purpose(nationalSecurity) \\ hasPartOf^{-}.DataUser(securityOfficer) \\ hasPartOf^{-}.Consent(\bot) \end{cases}$$

In stratum three, we deny the Taiwan national security officers to request disclosing the dataset `D`, where the personal citizenships are aliens, e.g., $\neg Taiwan$,

`Ab3` $\rightarrow$ `hasPartOf.Condition(Ab3)`. Therefore, the set of data can be disclosed under the Taiwan national security laws are data owners' consent are absent (or $\bot$) but the citizenships are Taiwan.

`hasPartOf.Condition(Ab3)`,

$$Ab3 = \begin{cases} hasPartOf.Condition(Ab2) \\ \cdots \\ hasPartOf^{-}.Nationality(\neg TW - citizenship) \end{cases}$$

The stratified $Datalog^{\neg}$ for policy exceptions handling to comply with both the privacy protection and the national security laws shown as follows:

`Request(?r)` $\wedge$ `hasCondition(?r, Ab1)` $\wedge$ `Condition(Ab1)`
$\wedge$ `DomainPolicy(?dmp)` $\wedge$ `hasCondition(?dmp, ?dmc)`
$\wedge$ `Condition(?dmc)` $\wedge$ `isSubsumed(Ab1, ?dmc)`
$\wedge$ `domainPolicyForSPD(?dmp, ?spd)`
$\longrightarrow$ `getInTo(?r, ?spd)` $\longleftarrow$ (7)

The dataset `A` by default is $\neg disclosed$ in the closed privacy data protection policy assumption. After the rule (2) replaced by the rule (7), followed by the entire rules' execution, e.g., from rule 1 to rule 6, in Section 8.3.1, we derive a disclosed dataset `B` $\subset$ `A`. The disclosed dataset `C` $\subset$ (`A` $-$ `B`) is created by replacing Ab1 with Ab2 in the rule (7). Finally, we derive the $\neg disclosed$ dataset `D` by replacing Ab1 with Ab3 in the rule 7. So the final legal disclosed dataset is `E` $\subset$ (`C` $-$ `D`) (see Figure 5).

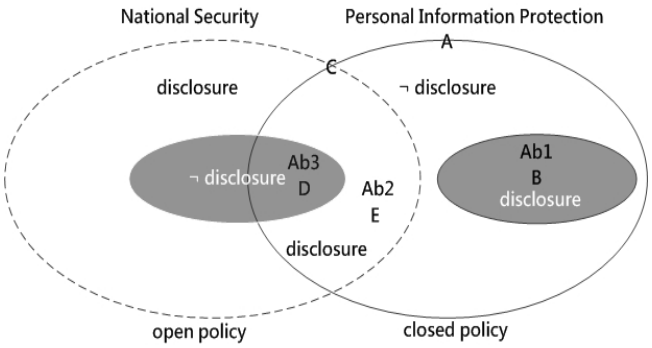

Figure 5: A policy's exceptions handling to disclose the dataset complied with both the privacy protection and the national security laws

9. CONCLUSION AND FUTURE WORK

We extend our previous work [25] [26] on a semantic privacy preserving model to provide legalized data integration and protection services in the semantic cloud. We propose a solution to overcome the privacy and legal obstacles when Cloud Service Providers (CSPs) intend to deploy their cloud resources and services for their potential customers. Semantic web technologies are applied to the legal policy representation in the semantic cloud for data integration and protection. The legal policies, represented as a combination of ontologies and stratified Datalog rules with negation, are used for a legal policy's exceptions handling through defeasible reasoning.

Legal policies are automatically enforced in the super-peer to enable Law-as-a-Service (LaaS) and subsequent queries for CSPs and their customers. The law-aware super-peer is a unique guardian, which provides data integration and protection services for its peers within a super-peer domain. Each super-peer enforces the legal policies to enable data integration and protection services. Finally, a scenario of a pandemic investigation scenario is demonstrated to explain how our LaaS provided by a super-peer is applicable for a dataset disclosure with a policy's exceptions handling.

Future work includes: exploring the defeasible reasoning of policy exceptions handling through a hybrid integration of DL-based ontologies and LP-based stratified Datalog rules proposed in [14], and furthermore exploiting the legal policy expressive power and computational complexity under this hybrid integration.

Acknowledgements

This research was partially supported by the NSC Taiwan under Grant No. NSC 100-2221-E-004-011-MY2.

10. REFERENCES

[1] M. I. Abbadi. Self-managed services conceptual model in trustworthy clouds' infrastructure. In *Workshop on Cryptography and Security in Clouds*, 2011.

[2] A. Barth et al. Privacy and contextual integrity: Framework and applications. In *IEEE Symposium on Security and Privacy*, 2006.

[3] D. Beneventano et al. Querying a super-peer in a schema-based super-peer network. In G. Moro et al., editors, *Databases, Information Systems, and Peer-to-Peer Computing*, LNSC, pages 13–25. Springer, 2007.

[4] A. Boer. *Legal Theory: Sources of Law and the Semantic Web*. IOS Press, 2009.

[5] A. P. Bonatti. Datalog for security, privacy and trust. In *Datalog 2010*, LNCS 6702, pages 21–36. Springer, 2011.

[6] S. Cabuk et al. Towards automated security policy enforcement in multi-tenant virtual data centers. *Journal of Computer Security*, 18:89–121, 2010.

[7] D. Calvanese et al. Data management in peer-to-peer data integration systems. *Global Data Management*, pages 177–201, 2006.

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

[9] S. Ceri et al. What you always wanted to know about Datalog (and never dared to ask). *IEEE Trans. on knowledge and data engineering*, 1(1), 1989.

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

[11] A. Datta et al. Understanding and protecting privacy: Formal semantics and principled audit mechanisms. In *7th International Conference on Information System Security*, 2011.

[12] I. Deyrup et al. Cloud computing and national security laws. Technical report, The Harvard Law National Security Research Group, 2010.

[13] A. Eberhart et al. Semantic technologies and cloud computing. In D. Fensel, editor, *Foundations for the Web of Information and Services*, pages 239–251. Springer, 2011.

[14] T. Eiter and G. Ianni. Rules and ontologies for the semantics web. In *Reasoning Web 2008*, LNCS 5224, pages 1–53. Springer, 2008.

[15] J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer, 2007.

[16] S. Foresti. *Preserving Privacy in Data Outsourcing*. Springer, 2011.

[17] 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.

[18] 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.

[19] F. T. Gordon. The legal knowledge interchange format (LKIF) ESTRELLA deliverable d4.1. Technical report, ESTRELLA, 2008.

[20] P. Haase et al. Semantic technologies for enterprise cloud management. In *International Semantic Web Conference 2010*, pages 98–113, 2010.

[21] A. Halevy et al. Schema mediation in peer data management systems. In *Proc. 19th Int. Conference on Data Engineering (ICDE)*, pages 505–516, 2003.

[22] A. Halevy et al. The Piazza peer data management system. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):787 – 798, july 2004.

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

[24] 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.

[25] Y. J. Hu, W. N. Wu, and J. J. Yang. Semantics-enabled policies for information sharing and protection in the cloud. In *Proc. of 3rd Int. Conf. on Social Informatics*, LNCS 6984, Oct. 2011.

[26] Y. J. Hu and J. J. Yang. A semantic privacy-preserving model for data sharing and integration. In *International Conference on Web Intelligence, Mining and Semantics (WIMS'11)*. ACM Press, May 2011.

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

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

[29] L. Lessig. *Code version 2.0*. Basic Books, 2006.

[30] J. Madhavan et al. Web-scale data integration: You can only afford to pay as you go. In *Proc. of CIDR-07*, 2007.

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

[32] J. W. Perry et al. *Protecting Individual Privacy in the Struggle Against Terrorists: A Framework for Program Assessment*. The National Academies Press, 2008.

[33] L. J. Pollock. Defeasible reasoning. In A. J. and L. Rips, editors, *Reasoning: Studies of Human Inference and its Foundations*. Cambridge University Press, 2008.

[34] R. Popp and J. Poindexter. Countering terrorism through information and privacy protection technologies. *IEEE Security & Privacy*, 4(6):24–33, 2006.

[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.

[36] J. D. Weitzner et al. Creating a policy-aware web: Discretionary, rule-based access for the world wide web. In E. Ferrari and B. Thuraisingham, editors, *Web and Information Security*, pages 1–31. IGI, 2006.