Future Internet **2012**, *4*, 929-954; doi:10.3390/fi4040929

OPEN ACCESS

future internet

ISSN 1999-5903

www.mdpi.com/journal/futureinternet

Article

Semantic Legal Policies for Data Exchange and Protection across Super-Peer Domains in the Cloud

Yuh-Jong Hu *, Win-Nan Wu, Kua-Ping Cheng and Ya-Ling Huang

Emerging Network Technologies (ENT) Lab, Department of Computer Science, National Chengchi University, Wen-Shan District, Taipei 11605, Taiwan; E-Mails: d9905@cs.nccu.edu.tw (W.N.W.); 99753025@nccu.edu.tw (K.P.C.); 99753026@nccu.edu.tw (Y.L.H.)

* Author to whom correspondence should be addressed; E-Mail: hu@cs.nccu.edu.tw; Tel.: +886-2-2938-7620.

Received: 21 September 2012; in revised form: 13 October 2012 / Accepted: 17 October 2012 / Published: 25 October 2012

Abstract: In semantic policy infrastructure, a Trusted Legal Domain (TLD), designated as a Super-Peer Domain (SPD), is a legal cage model used to circumscribe the legal virtual boundary of data disclosure and usage in the cloud. Semantic legal policies in compliance with the law are enforced at the super-peer within an SPD to enable Law-as-a-Service (LaaS) for cloud service providers. In addition, cloud users could query fragmented but protected outsourcing cloud data from a law-aware super-peer, where each query is also compliant with the law. Semantic legal policies are logic-based formal policies, which are shown to be a combination of OWL-DL ontologies and stratified Datalog rules with negation, *i.e.*, so-called non-monotonic cq-programs, for policy representation and enforcement. An agent at the super-peer is a unique law-aware guardian that provides protected data integration services for its peers within an SPD. Furthermore, agents at the super-peers specify how law-compliant legal policies are unified with each other to provide protected data exchange services across SPDs in the semantic data cloud.

Keywords: semantic data cloud; semantic data exchange; Law-as-a-Service (LaaS); semantic legal policies; privacy protection

---

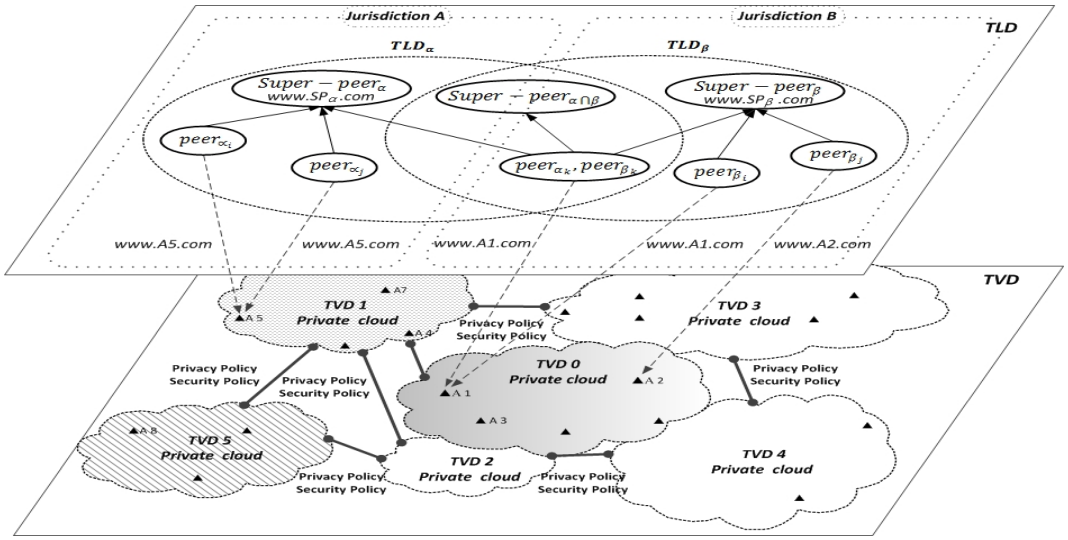
1. Introduction

Cloud computing has become a generic term that describes 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). A new, spectacular phenomenon of data sharing and service integration is possible within the cloud computing environment. This paves the way for technological innovation when applying semantic technologies to cloud computing [1].

Current cloud infrastructures do not provide enough self-managed services for their cloud users. Therefore, a cloud provider's internal employees must use manual service management, which requires intensive human intervention to explore and allocate available virtual resources for cloud users [2]. This is certainly not adequate when cloud resources are agile and deployed over the wide-scale of the Internet. We need self-managed Software-as-a-Service (SaaS) not only for the automatic allocation of various cloud available resources but also to enforce security and privacy policies without too much human intervention. *Law-as-a-Service (LaaS)* further enhances automatic security and privacy policy to provide law-aware semantic policy enforcement in the cloud [3].

In this paper, relational structured data are re-modeled as ontologies and used for data integration and exchange (see Figure 1). This leverages the abstract concept representation and reasoning of ontologies, which do not exist in the relational database [4]. Furthermore, stratified Datalog rules with exceptions handling extends ontologies to empower data protection and query services [5]. We have built a semantic data cloud for data integration and exchange. Furthermore, we have also applied Semantic Web technology to represent and enforce semantic legal policies for data protection in the cloud [6].

Figure 1. Law-aware semantic policy infrastructure, where a TLD is a legal cage, designated as an SPD, to circumscribe the legal virtual boundary of data disclosure and usage. A TVD is a logical cage that provides security and privacy policies, corresponding to semantic legal policies.



1.1. Semantic Policy Infrastructure

Current data protection laws are not up-to-date on handling data sharing and protection in the semantic cloud. We address the associated 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.

We propose a law-aware semantic policy infrastructure to provide LaaS for various cloud service providers (CSPs) and their potential customers. In this paper, we extend our previous work [7], where a Super-Peer Domain (SPD) for modeling a specific Trusted Legal Domain (TLD) enables data integration in the semantic policy architecture. This major revised version is explicitly different in two directions from the previous one we published in the WIMS'12 conference. (1) We introduce semantic data integration from outsourcing data in terms of the fragmentation of sensitive relationship to prevent curious but honest cloud service providers from using the data. Furthermore, we address the semantic data exchange between super-peers. A super-peer is in charge of an SPD, which is corresponding to a virtual private cloud (see Figure 1); (2) Policy exception is handled by cq-programs non-monotonic reasoning in $SPD_{\alpha \cap \beta}$.

Similar to a privacy appliance [8], an agent in the super-peer is a unique law-aware guardian that provides LaaS to its peers. The super-peer is also a trusted proxy of an SPD that provides a query interface between its peers and a user. Peers own real fragmented data sources that are directly mapped from relational database tables. Therefore, the super-peer provides a data integration service to its peers. Furthermore, the super-peer specifies how law-compliant semantic legal policies that are unified and enforced among SPDs to achieve data exchange. The enforcement of unified semantic legal policies not only protects integrated data from a super-peer's own peers but also protects exchanged data from another SPD.

1.2. Principles of Data Protection Laws

Regarding data protection legal issues, processing 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, to handle semantic legal policies in compliance with emerging data protection laws, we use flexible relationship mapping between TLD and TVD layers. This allows us to enable/disable security and privacy policies dynamically in the TVD layer using self-managed LaaS.

The principles of using privacy protection laws in the cloud depend on three criteria [9]:

- The registration principle: the location of service provider registration, which enables data collection services;
- The nationality principle: the nationality of the data owner whose data are being used;
- The territoriality principle: the data center location where the actual data processing occurs.

Currently, there is no consensus on which principles should be used for enforcing 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 to choose flexibly any principles of the privacy laws with which

they prefer to comply. We also ensure that all of the subsequent queries in the resources and services deployment cloud are compliant with the principles of selective laws. We manually unify semantic legal policies to avoid any possible conflicts of data disclosure and exchange across jurisdictions.

Whenever the laws used for regulating cloud computing are updated and expected to agree with the laws of different jurisdictions, semantic legal policies, modeled as a combination of revised ontologies and rules, are re-mapped to the updated security and data policies in Trusted Virtual Domains (TVDs). A TVD consists of a set of distributed virtual machines (VMs), storage for the VMs, and a communication medium interconnecting the VMs in the OpenTC [3]. A semantic cloud of TLDs is established over the OpenTC TVDs (see Figure 1). Therefore, we ensure that our law-aware semantic cloud policies are always compliant with the most up-to-date laws for cloud operations.

1.3. Research Issues and Contributions

1.3.1. Research Issues

We identify several research issues in this study (i) how to empower semantic technologies for cloud computing to establish law-aware semantic cloud policies; (ii) how to achieve data integration and exchange after data are fragmented for protection in an outsourcing SPD; (iii) how to use semantic legal policies to represent and interpret laws, especially for data protection and national security laws and to further ensure the legality of data exchange and access across jurisdictions; (iv) how to unify semantic policies to allow defeasible (or non-monotonic) reasoning of a policy's exceptions handling.

In this study, we use non-monotonic reasoning, in the form of default logic, instead of defeasible description logic because the high computational complexity of defeasible reasoning for description logic. Defeasible reasoning from the philosophical literature was rediscovered in artificial intelligence as non-monotonic reasoning, attempting to solve the "frame problem" [10]. Defeasible reasoning differs from deductive reasoning in that the reason-schemes employed in defeasible reasoning can have defeaters.

1.3.2. Contributions

Our main contributions are (i) the establishment of a law-aware semantic cloud policy infrastructure to verify the feasibility of LaaS concepts; (ii) the design and enforcement of semantic legal policies using fragmented outsourcing data for data integration and protection in an SPD of single jurisdiction; (iii) the unification of semantic legal policies from multiple SPDs for data exchange and protection across jurisdictions. Finally, we exploit policy's exceptions handling by default logic in cq-programs to support non-monotonic reasoning for description logics [11].

1.3.3. Outline

This paper is organized as follows. In Section 2, we first introduce the background. In Section 3, we present a law-aware semantic data cloud. In Section 4, we address the issues of modeling TLD(s) for semantic legal policy enforcement. Semantic legal policies are formally defined in Section 5. In Section 6, we unify two types of semantic legal policies to address the problem of exceptions handling.

In Section 7, we present semantic legal policy enforcement, focusing on non-monotonic reasoning of a policy's exceptions handling. In Section 8, related studies are presented. We conclude this paper and note possible future work in Section 9.

2. Background

2.1. A Super-Peer Domain Model

A Peer Data Management Systems (PDMS), such as PAYGO and Piazza, were demonstrated as the best way to achieve wide-scale data integration over the Internet [12–14]. However, the PAYGO and Piazza systems only use a relational data model that hampers our use of a conceptual-based for information sharing. Moreover, it is difficult to enact data sharing in a pure peer data integration architecture because of the difficulty of describing the nature of relationships among many unstructured peers. It is certainly a great challenge to provide unifying semantic legal policy services for effective data integration and protection in an unstructured peer data management system.

We propose an SPD model to allow for data integration and protection in a jurisdiction and furthermore to enable data exchange across jurisdictions. Within an SPD, a super-peer specifies its semantic legal policies based on a type of law to regulate a jurisdiction. Any peer registers at a super-peer, pledging to comply with a law declared as semantic legal policy in a super-peer for data integration. We also allow a super-peer to exchange data with another super-peer. This implies that when peers are affiliated with different super-peers, the semantic legal policies declared in these super-peers are unified to enact data exchange between these peers (see Section 6).

2.2. Queries as Views

In terms of the data integration of multiple data sources, three approaches have been proposed to model a set of *source descriptions* that specify the semantic mapping between the source and global schema [15]. 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 addresses the case in which a 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 the ontology and the dynamic data sources, are established by defining each concept in the data sources as a view over the global schema [4,16]. 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].

In this paper, on the one hand, data integration uses LAV and GAV mappings within an SPD to reformulate a user query into a query over the source schemas. On the other hand, data exchange between super-peers uses GLAV mappings between different SPDs. More specifically, in a data exchange setting

a tuple generating dependency (tgd) for a set of source-to-target dependencies or an equality-generating dependency (egd) for target dependencies is extended to Datalog shown as a GLAV mapping [18]. However, data integration and exchange are hampered by legitimate and widespread privacy concerns. We need a technique that enables data integration and exchange without losing a user's privacy [19,20]. An acronym table for frequently used terms are shown as Table 1.

Table 1. An acronym table for frequently used terms.

Acronym	Full spelling	Acronym	Full spelling
TLD	Trusted Legal Domain	TVD	Trusted Virtual Domain
SPD	Super-Peer Domain	sp	super-peer
GAV	Global-As-View	XaaS	Everything-as-a-Service
GLAV	Global-Local-As-View	LAV	Local-As-View
CQ	Conjunctive Query	VMs	Virtual Machines

2.3. Stratified Datalog¬ for Non-Monotonic Reasoning

Datalog is a database query language based on the logic programming paradigm: a set of ground facts, called the *Extensional Database* (EDB), is physically stored in a relational database, and a Datalog program P is called the *Intensional Database* (IDB). A Datalog program P is a mapping from EDB-facts to IDB-facts. Stratified *Datalog*¬, which reduces data complexity and offers non-monotonic reasoning, is an extension of pure Datalog with rule stratification and negation [21].

2.4. Conjunctive Query Programs (CQ-Programs)

The integration of ontologies and rules can be classified as heterogeneous or homogeneous [22]. Heterogeneous integration for hybrid reasoning is further distinguished as being loosely or tightly coupled. Description Logic Program (DLP) and Semantic Web Rule Language (SWRL) are examples of homogeneous integration, but they lack non-monotonic reasoning capabilities for policy exceptions handling. Therefore, we adopt one of the loosely-coupled approaches, *i.e.*, conjunctive query programs (cq-programs), which is an extended version of a description logic program (dl-program), to design and enforce semantic legal policies [23].

The semantic legal policies are expressed as a *cq-program*, e.g., a pair $(\mathcal{T}, \mathcal{P})$, where $\mathcal{T}$ is the DL-based ontology and $\mathcal{P}$ consists of a finite set of non-monotonic datalog rules. A *cq-rule* has the form:

$$a \longleftarrow b_1, ..., b_n, not\ \ b_{n+1}, ..., not\ \ b_m \tag{1}$$

where a is a rule predicate and any $b_1, ..., b_m$ may be a DL predicate or a rule predicate.

The cq-program combines datalog rules with negation under stable model (or answer set) semantics with OWL DL. The difference between stable model semantics and well-founded semantics is that between two-valued (true or false) and three-valued (true, false, or unknown) logic. In practice, the two semantics coincide in the stratified logic program. Negation-as-failure (NAF) for weak negation ($\sim$) in the closed world assumption (CWA) and explicitly negative knowledge for strong negation ($\neg$)

are allowed in the cq-program. In fact, Reiter-style default logic and CWA can be implemented in cq-programs to support non-monotonic reasoning of description logics [11].

2.5. Prioritized Default Theory

Let $\Delta = (\mathcal{T}, \mathcal{D}, \prec)$ be a prioritized default theory, where $\mathcal{T}$ is the DL-based ontology in the *cq-program* of $(\mathcal{T}, \mathcal{P})$ and $\mathcal{D} = \{\delta_0, \cdots, \delta_n\}$ is a finite set of defaults with strict priorities. $\mathcal{P}$ consists of a finite set of non-monotonic datalog rules [24,25]. The *normal* default $\delta = \frac{\varphi:\psi}{\psi}$ is sufficient to model our exceptions, where φ is *prerequisite*, ψ is *justifications*, and ψ the *consequent* of δ in $\mathcal{D}$. We apply the novel transformation Ω of default theories into cq-programs, which is based on the *select-default-and-check* principle. The evaluating extensions principle in default theories is: "If the prerequisites can be derived, and the justifications can be consistently assumed, then the conclusion can be concluded [23]".

3. Law-Aware Semantic Cloud

A policy-aware infrastructure gives users greater transparency in their online interactions and helps both people and machines "play by the rules" relevant to social interactions [26]. We intend to achieve similar objectives. However, we focus more on the issue of enforcing law-aware policies in the semantic data cloud to fulfill two visions:

1. The semantic data cloud 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 infractions when they plan to deploy their resources and services. LaaS also provides CSPs with transparent updating semantic policies that are compliant with the most up-to-date laws.
2. Ontologies and stratified Datalog rules with negation are used for representing semantic legal policies to enable query services for real cloud end-users. Semantic legal policies are manually unified but automatically enforced by the systems because metadata extracted from the semantic data cloud are used in deciding whether the integrated data satisfy the relevant legal policy's preconditions. If the data usage context satisfies the preconditions, data are *disclosed*. Otherwise, they are hidden (or $\neg$*disclosed*).

3.1. A Pandemic Investigation Scenario

Example 3.1. 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, where authorizations are denied by default based on the registration and nationality principles described in Section 1.2. The first policy exception, Ab1, states that no personal data should be disclosed unless a data owner's prior consent is obtained.

Whenever a data disclosure request comes from a data owner's national security officer, the *open policies* of the corresponding national security laws are applied, where authorizations are granted by default. As long as the officer follows legal procedure in supporting plausible evidence, this request is granted without the data owners' consent. However, any national security officer cannot request an

alien's personal data unless the data owner's prior consent was obtained. The second policy exception, Ab2, occurs in this situation.

The α airline Inc. pledges to follow Singapore's data protection laws but allows data disclosure when any national security officer requests his/her own citizen's flight information. An SPD_α is created for the α airline in compliance with Singapore's data protection laws in TLD_α, where queries request data from the domain's distributed data centers. Another SPD_β is created for national security officers of Taiwan CDC to enforce national security laws for a pandemic investigation, based on territoriality and nationality principles. A data disclosure exception, Ab2, is used for national security law enforcement when citizens are not Taiwanese nationals. Thus, these data cannot be disclosed unless a data owner's prior consent was obtained. During the recent H1N1 pandemic period, a national security officer in Taiwan tried to trace the original H1N1 carriers who possibly took inbound flights from foreign territories, including Singapore, to Taiwan within the preceding 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 either subject- or pattern-based queries ("Subject-based queries allow data users to query a specific person's complete profile, while pattern-based queries allow data users to adopt a predicate model to create specific features that correspond to anticipated targets and find people with these features in the information space without disclosing their complete profile [8]".) are initiated at a different super-peer?

4. Semantic Super-Peer Data Cloud

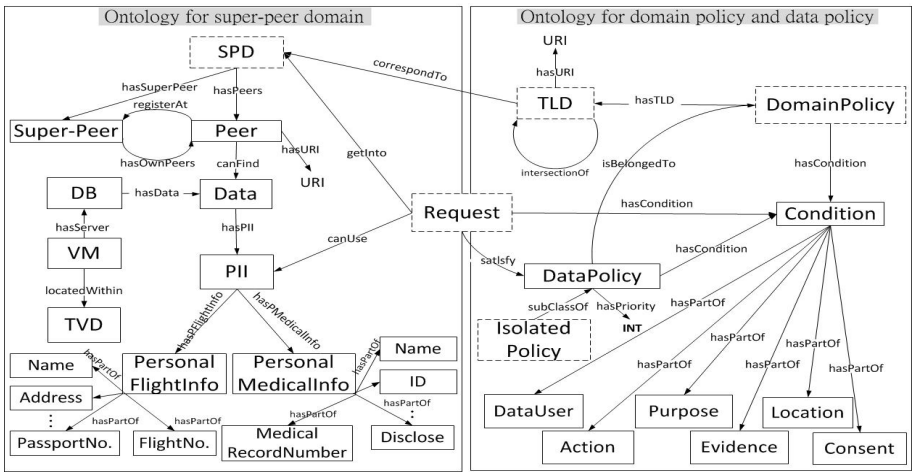
In a previous study [27], unstructured data integration is too complex for heterogeneous peer schemas when the number of peers is large. In a super-peer network architecture, we group a set of peers into a super-peer domain and organize them into a two-level architecture as in a previous work [28], where the lower level is called the peer level and the upper level is called the super-peer level.

More precisely, a peer transforms relational data sources in a TVD into a local ontology in a TLD. An agent in the super-peer is a guardian of a data integration system; this agent integrates all of its local peers' ontologies into a global ontology through ontology mapping, alignment, and merging [29,30].

The establishment of emergent semantics in a super-peer data cloud allows for flexible data integration from another SPD 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 exchange without generating semantic ambiguity in ontology mappings among super-peers.

The semantics of a super-peer data cloud is described as the policy ontology, including the modular concepts of SPD, domain policy, and data policy (see Figure 2). Any peer from an SPD can contribute new data, schema, or even mappings through its super-peer to other SPDs for data exchange. We define a super-peer data cloud system as a set of SPDs $\Pi = \{\pi_1, ..., \pi_n\}$, where each π_i corresponds to a TLD. It is an autonomous information site that exports its data in terms of a super-peer schema to another SPD.

Figure 2. The semantics of a super-peer data cloud is described as the policy ontology, which includes two modular concepts (**a**) SPD; (**b**) domain and data policy. An SPD 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 for querying data from the super-peer.



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

4.1. Semantics of a TLD

In an SPD π_α, actual data are stored in a set of fragmented relational data sources $DS_\alpha = \{ds_1, ..., ds_m\}$ of a database. In an outsourcing database, the sensitive relationships (or properties) of the attributes in the tables are identified and segmented into fragmented data sources to ensure the data protection criteria are satisfied [31]. Using GAV local mappings, we associate a set of local peer $P_\alpha = \{peer_1, ..., peer_n\}$ 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 a super-peer global view.

An SPD $\pi_\alpha \in \Pi$ can be defined as $(P_\alpha, SPD_\alpha, GS_\alpha, LS_{peer_i}, M_\alpha, DS_\alpha)$:

- An $sp_\alpha \in SPD_\alpha$ is the only node in an SPD π_α, which allows an $agent_\alpha$ to enforce semantic legal policies. This enforcement empowers $agent_\alpha$ in an sp_α to facilitate information collection through a conjunctive query $CQ_{\pi_\alpha}(sp_\alpha)$ posed to the global schema GS_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 [32].
- Through local LAV mapping assertions, a global schema GS_α provides an integrated view for a set of peers from P_α in π_α. We propose 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 an sp_α for $peer_i$.

- A set of peers from P_α are mediators. $Peer_i \in P_\alpha$ maps its local ontology schema, LS_{peer_i}, to a set of fragmented relational data sources, ds_i, from DS_α in π_α. Therefore, a query uses 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 $peer_i$ that maps to the SQL of $CQ_{\pi_\alpha}(DS_\alpha)$ over a set of fragmented 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 sp_α and a set of peers from P_α in π_α. The semantics of a set of global mapping assertions created from a Datalog rule language among super-peers are addressed in Section 4.2.
- A set of local data sources, ds_i, from DS_α, are fragmented relational structured data that store the materialized instances.

4.2. Semantics of Multiple TLDs

When LaaS supports cloud resource deployment and queries across TLDs, the laws declared in each TLD are unified to comply with all TLDs. An SPD π_α of TLD_α is related to another SPD π_β of TLD_β using a set of super-peer GLAV semantic mapping assertions. A super-peer semantic schema mapping assertion between TLD_α and TLD_β is expressed as follows:

$$CQ_{\pi_\alpha}(sp_\alpha) \rightsquigarrow CQ_{\pi_\beta}(sp_\beta) \tag{2}$$

where $CQ_{\pi_\alpha}(sp_\alpha)$ is a source conjunctive query over the sp_α in an SPD $\pi_\alpha \in \Pi$; and $CQ_{\pi_\beta}(sp_\beta)$ is a target conjunctive query over the sp_β in an SPD $\pi_\beta \in \Pi$. A $CQ_{\pi_\alpha}(sp_\alpha)$ is defined as an authorized legal view of an SPD π_α whenever the sp_α intends to export its data by unifying its semantic legal policies with another SPD π_β. The global schema GS_α of sp_α is mapped to another sp_β's global schema GS_β by the super-peers' GLAV semantic mapping assertions.

When queries go through the intersection of TLDs across *law-aware* super-peers, we manually unify the pre-arranged semantic legal policies to discover mapping assertions from the vocabulary of sp_α's global ontology schema to the vocabulary of sp_β's global ontology schema. Furthermore, potential policy conflicts between these unifiable semantic legal policies should be resolved with Datalog rules by policy exceptions handling. A semantic legal policy's exceptions are handled by non-monotonic reasoning with stratified $Datalog^\neg$ rules, as shown in Section 7.4.

4.3. Semantic Data Exchange Between SPs

Semantic data exchange between super-peers is the problem of taking a data structure under a source schema of a TLD_α and creating an instance of a target schema of a TLD_β that reflects the source data as accurately as possible. A semantic data exchange setting $(S_\alpha, T_\beta, \sum_{st}, \sum_t)$ consists of a source schema $\mathtt{S}_\alpha$, a target schema $\mathtt{T}_\beta$, a set of source-to-target dependencies $\sum_{st}$, and a set of target dependencies $\sum_t$. $\sum_{st}$ is a *tuple-generating dependency* (tgd). This is a super-peer semantic schema mapping assertion between TLD_α and TLD_β, described as Formula (2) in Section 4.2. Moreover, each target dependency in $\sum_t$ is either a *tuple-generating dependency* (tgd) or an *equality-generating dependency* (egd) [18]. Let $\sum$ be a set of tgds over a fixed schema. A set of tgds is *weakly acyclic* if the dependency graph has

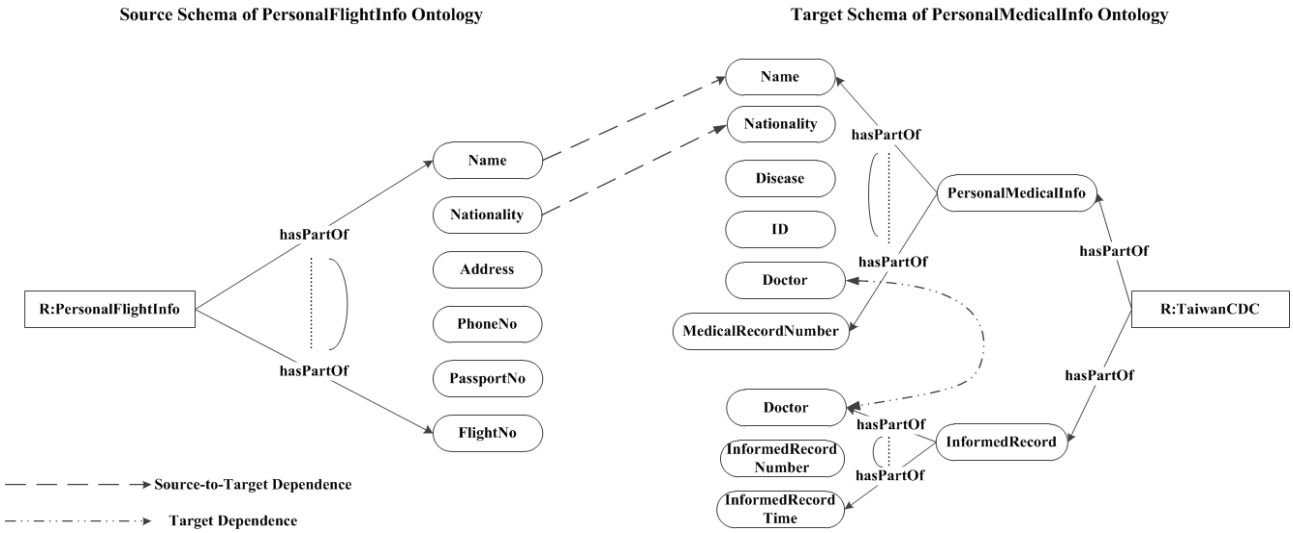
no cycle going through dot-line special edges (see Figures 3 and 4). This guarantees that the chase (or query) from target and source schemas will be terminated in polynomial time.

One of the source-to-target dependencies $\sum_{st}$ from the source schema of personal flight information to the target schema of personal medical information is described as follows (see Figure 3):

A source-to-target dependency:

$$\begin{aligned}\sum_{st} = \{&\texttt{PersonalFlightInfo}(?\texttt{n}, ?\texttt{na}, ?\texttt{a}, ?\texttt{p}, ?\texttt{pa}, ?\texttt{f}) \longrightarrow \\ &\exists \texttt{D} \exists \texttt{I} \exists \texttt{D}_\texttt{o} \exists \texttt{M}.\texttt{PersonalMedicalInfo}(?\texttt{n}, ?\texttt{na}, \texttt{D}, \texttt{I}, \texttt{D}_\texttt{0}, \texttt{M}) \\ &\wedge \exists \texttt{IRN} \exists \texttt{IRT}.\texttt{InformedRecord}(\texttt{D}_\texttt{o}, \texttt{IRN}, \texttt{IRT})\}\end{aligned} \tag{3}$$

Figure 3. A source-to-target dependency $\sum_{st}$ is defined from the source schema of personal flight information to the target schema of medical information.

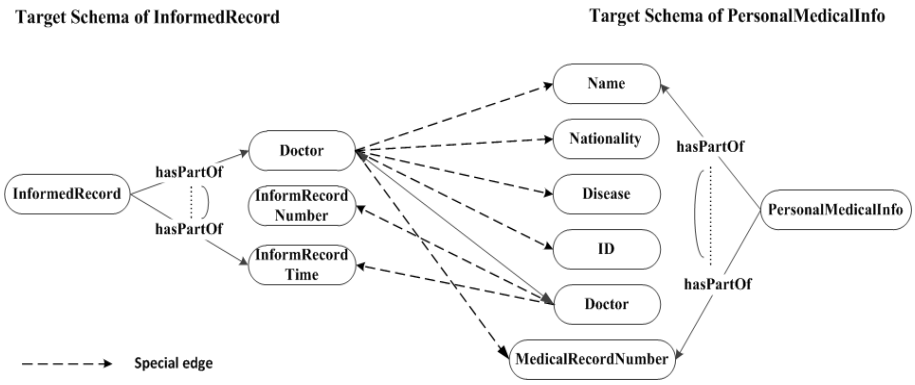


A target dependency:

$$\begin{aligned}\sum_{t} = \{&\texttt{InformedRecord}(?\texttt{do}, ?\texttt{irn}, ?\texttt{irt}) \longrightarrow \\ &\exists \texttt{N} \exists \texttt{N}_\texttt{a} \exists \texttt{D} \exists \texttt{I} \exists \texttt{M}.\texttt{PersonalMedicalInfo}(\texttt{N}, \texttt{Na}, \texttt{D}, \texttt{I}, ?\texttt{do}, \texttt{M}), \\ &\texttt{PersonalMedicalInfo}(?\texttt{n}, ?\texttt{na}, ?\texttt{d}, ?\texttt{i}, ?\texttt{do}, ?\texttt{m}) \longrightarrow \\ &\exists \texttt{IRN} \exists \texttt{IRT}.\texttt{InformedRecord}(?\texttt{do}, \texttt{IRN}, \texttt{IRT})\}\end{aligned} \tag{4}$$

A set of weakly acyclic target dependencies $\sum_{t}$ from the target schema of InformedRecord to the target schema of PersonalMedicalInfo is shown as (see Figure 4):

Figure 4. A target dependency $\sum_t$ is defined from the target schema of InformedRecord to the target schema of PersonalMedicalInfo.



In Section 3.1, we have proposed a pandemic investigation scenario, in which Taiwan CDC officers enforced national security laws to trace the original source of H1N1 carriers, who possibly took inbound flights from a recent pandemic area. We first query the target schema of a personal medical information ontology using Formula (4). It is weakly acyclic; thus, the chase procedure for personal medical information with H1N1 disease will be terminated in polynomial time. Then, we query the instances of personal flight information at the source schema through the source-to-target dependency $\sum_{st}$ described by Formula (3). Semantic data exchange services ensure that both semantic data interoperability and law-compliant criteria are satisfied at the virtual legal domain created for the super-peer $sp_{\alpha\cap\beta}$, where a law-aware guardian $agent_{\alpha\cap\beta}$ at the $sp_{\alpha\cap\beta}$ is empowered by unifying the semantic legal policies offered by $agent_{\beta}$ and $agent_{\alpha}$ from their respective sp_{β} and sp_{α} nodes. For more details see Section 6.

5. Semantic Legal Policies

5.1. Legal Policy Representation

A formal (semantic) *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 expressed as a combination of ontology and rule languages. A *legal policy* is composed of ontologies and rules, where ontologies are created from an ontology language to express the domain concepts of a policy and rules are created from a rule language to express the enforcement of a policy.

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 expressed as rules. The privacy policy model used for access control in enterprises has been extensively investigated [33], where only Logic Program (LP)-based Datalog was used to design the privacy policy. A *global policy schema* allows for data integration by unifying *regular policies* from a variety of structured data sources, where the *global policy schema* includes integrated ontologies and rules.

When rules specified as stratified Datalog with negation are used for non-monotonic reasoning rules, the research challenge is determining how to integrate two families of logic *i.e.*, DL and LP, for semantic

legal policy enforcement under a non-monotonic semantics. Expressiveness is not the only issue because hybrid integration usually involves high computational complexity. It is also important to ensure the appropriate hybrid integration of ontologies and rules to design policy languages for privacy protection policies. Unfortunately, this issue has not yet been completely studied [5].

5.2. Legal Policy Compliance

Legal policy compliance addresses the issues of data access and service execution in the semantic data cloud. Semantic legal policy enforcement should satisfy up-to-date laws within a jurisdiction. However, resources, data, and services in the cloud are usually dispersed throughout the Internet. Anyone, if authorized, should be allowed to request anything from anywhere at any time. In this case, we might have to regulate a data request by unifying laws across jurisdictions. This raises the regulation compliance issue regarding how to ensure that semantic legal policies, which satisfy the data usage context, are correctly enforced.

A *data usage context* is created for each user. It is a precondition when applying laws for a query in a TLD. In the policy ontology described in Figure 2 in Section 4, whenever the concept of a data usage context is subsumed by a domain policy's context, this data request enters a specific TLD. We comply with the laws of a *domain policy* because the subsumption criteria of a data usage context are satisfied. After a domain policy is chosen, an applicable *data policy* belonging to this TLD is initiated to enable real data access. However, this data usage is only used for a single TLD.

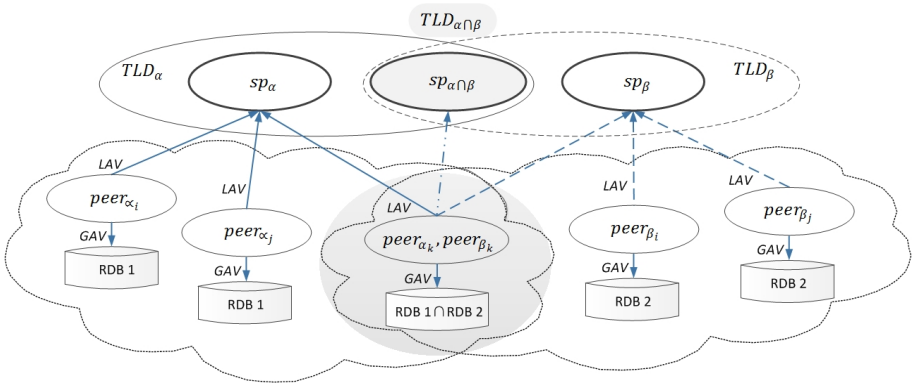
When data are used across jurisdictions, such as at the intersection of data protection and national security TLDs, we need an iterative legal policy enforcement process to achieve the selective revelation of anonymous personal identifiable information (PII). This is a pattern-based query, which is only allowed at the intersection of two SPDs. It has been unusually challenging to build a legal framework for protecting individual privacy in the struggle against terrorists since the 9-11 terrorist attack on the U.S. [8,34]. The wide distribution of cloud computing services will certainly exacerbate this challenge. We attempt to address this research issue and provide one possible solution based on semantic legal policy enforcement.

6. Unifying Semantic Legal Policies

We propose a semantic legal policy framework to serve flexible policy deployment, integration, and enforcement. In this policy framework, semantic legal policies representing privacy protection law α and national security law β are unified at the $sp_{\alpha\cap\beta}$ of $TLD_{\alpha\cap\beta}$, where an SPD of $TLD_{\alpha\cap\beta}$ is at the intersection of TLD_{α} and TLD_{β} jurisdiction (see Figure 5).

Regarding national security law, the Electronic Communication Privacy Act (ECPA) was enacted in 1986 and did not foresee the proliferation of cloud computing [35]. Therefore, we need to reform the ECPA to balance personal privacy rights and government law enforcement needs within the cloud computing environment.

Figure 5. 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 control in its SPD. For example, $agent_{\alpha}$ at the sp_{α} uses the data protection law in TLD_{α} and $agent_{\beta}$ at the sp_{β} uses the national security law in TLD_{β}. Finally, a query for data exchange, abode by $agent_{\alpha\cap\beta}$, is by unifying laws at the $sp_{\alpha\cap\beta}$ in $TLD_{\alpha\cap\beta}$.



6.1. A Peer Registers at a TLD

A peer p_i should register at the sp_{α} before this peer can offer its data for integration in an SPD_{α}, This registration action implies that p_i pledges to obey the privacy protection law by applying specifications of the data usage context declared in the sp_{α}.

Based on this data usage context specification, p_i uses the LAV source description to export its data to sp_{α} for data integration. Peer p_i also registers at another 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 privacy projection and national security legal policies, respectively, are unified and enforced after collecting data from p_i.

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

One possible solution to this legal discrepancy is to enact a service level agreement (SLA) between owners of a database and a data center before the database is installed in the data center's VMs. The SLA provides the necessary information for a database owner to ensure that he/she is aware of this legal domain discrepancy. Another possible solution for preserving privacy in data outsourcing is to enforce privacy over data collections by combining *data fragmentation* with *encryption* to avoid illegal data usage requests from *curious but honest* cloud providers [31]. For more details, see Section 7.1. Unless national security law enforcement officers comply with the SLA and national security laws, any data disclosure request made without a data owner's prior consent will be rejected.

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

6.2. Query at the $TLD_{\alpha\cap\beta}$ for Data Exchange

In Figure 5, an $agent_\alpha$ in TLD_α enforces privacy protection law, and an $agent_\beta$ in TLD_β enforces national security law. When a data usage context 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 falls on national security; then, we enter the $TLD_{\alpha\cap\beta}$ legal domain for data exchange. We model the enforcement of national security law as the privacy policy's exceptions. Whenever a national security officer queries data at the $sp_{\alpha\cap\beta}$, the nationality principle shown in Section 1.2 allows another jurisdiction's privacy protection law to bend. However, only the anonymous PII are disclosed because we still have to somehow ensure that 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 privacy protection law and national security law to demonstrate how data are collected from peers who have been registered at the $sp_{\alpha\cap\beta}$. Two types of queries are available subject- and pattern-based queries, where a subject-based query allows us to access a specific data owner's complete profile. Conversely, a pattern-based query does not have specific access targets, so only data that satisfy pre-defined filtering conditions are disclosed.

At the $sp_{\alpha\cap\beta}$, we only provide pattern-based queries. This is in contrast with the queries provided at the sp_α and sp_β, where we provide both. Similar to the privacy appliance [8], a trusted $agent_{\alpha\cap\beta}$ at the $sp_{\alpha\cap\beta}$ is a guardian, who follows the laws and provides privacy protection and national security legal services while disclosing data from its registered peers within $TLD_{\alpha\cap\beta}$.

In summary, we manually unify privacy protection legal policies with national security legal policies at the intersection of TLDs while enforcing data exchange. This is not only to ensure privacy but also to encourage sharing data for national security purposes without the fear of privacy rights being violated.

7. Semantic Legal Policy Enforcement

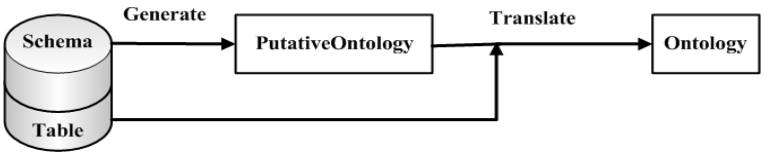
7.1. Semantic Data Outsourcing in an SPD

In this section, we demonstrate direct mapping from fragmented relational database tables to modular sub-ontologies in an outsourcing semantic data cloud. This prevents an illegal data request from *curious but honest* cloud providers. Whenever a data request is permitted by the LaaS, we ensure that this request is satisfied with the data protection criteria declared by data owners.

The relational database tables are first normalized using ordinary database techniques, e.g., first normal form (1NF), second normal form (2NF) [36]. The normalization technique reduces the number of duplicated tuples in the table. Then, the relational database tables are fragmented and mapped into

a putative ontology (see Figure 6). In Figure 7, tables in a SQL schema are directly mapped to the Semantic Web's putative ontology. The fragmented putative ontology of medical information is shown as a combination of modular sub-ontologies that are created from the relational database fragmented tables (see Figure 8).

Figure 6. A putative ontology is generated from relational database tables.



At the sp_{α} of an SPD_{α}, we provide a data request service through the semantic data integration of modular sub-ontologies [30]. Once the LaaS verifies this request and grants a permission, a link ontology is used to integrate the modular sub-ontologies. Semantic reasoning is performed from a link ontology to rediscover the sensitive relationships from previous modularized but fragmented sub-ontologies. Finally, a semantic data exchange service is provided by the guardian $agent_{\alpha \cap \beta}$ to achieve data exchange and protection across SPDs (see Section 6.2).

Figure 7. Direct mapping from the SQL schema of relational database tables to an OWL-based putative ontology.

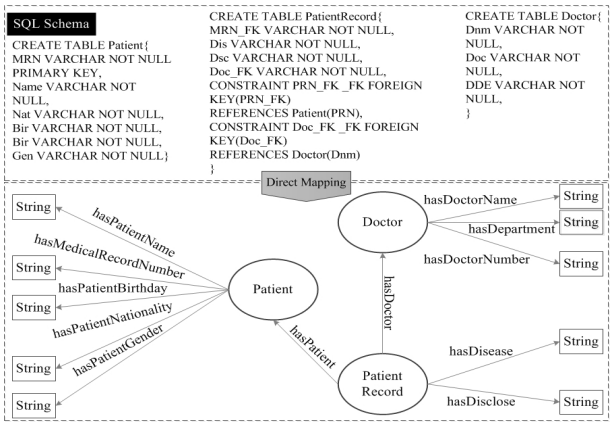
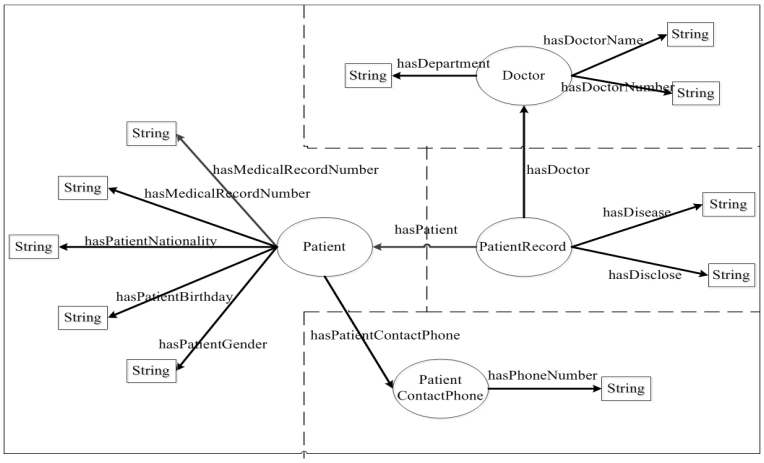


Figure 8. A modularized and fragmented putative ontology for personal medical information.



7.2. Legal Reasoning in SPD_α

A *legal policy* $(\mathcal{T}, \mathcal{P})$ is composed of ontologies $\mathcal{T}$ and rules $\mathcal{P}$, where ontology predicates are exported to the rules with the namespace indicator t to declare the original ontological predicate source. However, each rule still has its own predicates with the namespace indicator p. Based on the policy ontology described in Section 4, when a data request ?r with data usage context ?c satisfies a DomainPolicy(?dmp)'s data usage context ?dmc, this data request from a data user is allowed to enter the TLD(?tld) and enforce a data policy with operations on the PII dataset (see Rules 5 and 6).

In the pandemic investigation scenario presented in Section 3.1, under normal conditions, we enter sp_α in the SPD_α for subject queries as follows:

• A partial ontology for a domain policy:

$$\begin{aligned}
&\texttt{hasTLD.DomainPolicy(dmp)},\\
&\texttt{hasTLD}^{-}\texttt{.TLD(tld)}.
\end{aligned}$$

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

$$\begin{aligned}
&\texttt{hasCondition.DomainPolicy(dmp)},\\
&\texttt{hasCondition}^{-}\texttt{.Condition(dmc)}.\\
&\texttt{hasPartOf.Condition(dmc)},\\
&\texttt{hasPartOf}^{-}\texttt{.Purpose(checkIn)},\\
&\texttt{hasPartOf}^{-}\texttt{.DataUser(airlineStaff)},\\
&\texttt{hasPartOf}^{-}\texttt{.Action(read)}.\\
&\texttt{hasPartOf}^{-}\texttt{.Location(TW)},\\
&\texttt{hasPartOf}^{-}\texttt{.Consent}(\top).\\
&=1\,\texttt{hasSuperPeer}^{-}\texttt{.Super} - \texttt{Peer(sp)},\\
&\exists\,\texttt{hasPeers.Peer(p)},\\
&\forall\,\texttt{registerAt.Peer(p)},\\
&\exists\,\texttt{registerAt}^{-}\texttt{.Super} - \texttt{Peer(sp)}.
\end{aligned}$$

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

• Rules for a domain policy enforcement:

Rule (5) provides a concept link between an abstract TLD and a concrete SPD. In Rule (6), we determine whether a SPD should handle a data request based on this data request usage context, which is subsumed by a domain policy's context.

$$\begin{aligned}
&\texttt{t} : \texttt{DomainPolicy}(?\texttt{dmp}) \wedge \texttt{t} : \texttt{hasTLD}(?\texttt{dmp}, ?\texttt{tld})\\
&\wedge \texttt{t} : \texttt{correspondTo}(?\texttt{tld}, ?\texttt{spd}) \wedge \texttt{t} : \texttt{SPD}(?\texttt{spd})\\
&\longrightarrow \texttt{p} : \texttt{domainPolicyForSPD}(?\texttt{dmp}, ?\texttt{spd})
\end{aligned} \tag{5}$$

$$\begin{aligned} &\mathtt{t} : \mathtt{Request}(?\mathtt{r}) \wedge \mathtt{t} : \mathtt{hasCondition}(?\mathtt{r}, ?\mathtt{c}) \wedge \mathtt{t} : \mathtt{Condition}(?\mathtt{c}) \\ \wedge &\mathtt{t} : \mathtt{DomainPolicy}(?\mathtt{dmp}) \wedge \mathtt{t} : \mathtt{hasCondition}(?\mathtt{dmp}, ?\mathtt{dmc}) \wedge \mathtt{t} : \mathtt{Condition}(?\mathtt{dmc}) \\ \wedge &\mathbf{p} : \mathbf{isSubsumedByDefault}(?\mathbf{c}, ?\mathbf{dmc}) \wedge \mathtt{p} : \mathtt{domainPolicyForSPD}(?\mathtt{dmp}, ?\mathtt{spd}) \\ &\longrightarrow \mathtt{p} : \mathtt{getInTo}(?\mathtt{r}, ?\mathtt{spd}) \end{aligned} \tag{6}$$

In a predicate $\mathbf{p} : \mathbf{isSubsumedByDefault}(?\mathbf{c}, ?\mathbf{dmc})$ of Rule (6), the concept subsumption criteria is verified to determine whether a data request, with its structure condition attributes $?c$ is subsumed by the criteria of a domain policy context. In fact, each attribute is defined as a conceptual graph; therefore, the subsumption verification of each concept criterion is transformed into a conceptual graph-covering problem. This data request is granted only if the domain policy's conceptual graphs include the graphs of all of the request's attributes. Otherwise, it is rejected.

We do not address this issue further because the complex data structure of condition attributes must be modeled as function symbols for manipulation. However, the function symbols used in the datalog fragment usually introduces undecidable computation [11].

Instead, the default concept for condition ?c with any abnormal attribute subsumption will be verified through default logic to determine whether a data request with any abnormal condition Ab in ?c is subsumed (or defeasibly inherited) by the defaults in ?dmc within a domain policy. For more details about default reasoning, see Section 7.4. We allow a data request ?r using the PII ?pii of personal information as follows (see Rules (7–10)).

• A partial ontology for a data policy, which describes the concept of personal flight information available for user querying from the super-peer in an SPD:

satisfy.Request(r),
satisfy^{-}.DataPolicy(dap).

canFind.Peer(p),
canFind^{-}.PII(pii).

isBelongedTo.DataPolicy(dap),
isBelongedTo^{-}.DomainPolicy(dmp).

hasPII.Data(da),
hasPII^{-}.PII(pii),

hasPFlightInfo.PII(pii),
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:

$$\begin{gathered}\mathtt{t}:\mathtt{SPD}(?\mathtt{spd}) \wedge \mathtt{t}:\mathtt{hasSuperPeer}(?\mathtt{spd},?\mathtt{sp}) \wedge \mathtt{t}:\mathtt{Super}-\mathtt{Peer}(?\mathtt{sp}) \\ \wedge \mathtt{t}:\mathtt{hasPeers}(?\mathtt{spd},?\mathtt{p}) \wedge \mathtt{t}:\mathtt{Peer}(?\mathtt{p}) \wedge \mathtt{t}:\mathtt{registerAt}(?\mathtt{p},?\mathtt{sp}) \\ \longrightarrow \mathtt{p}:\mathtt{hasOwnPeers}(?\mathtt{sp},?\mathtt{p})\end{gathered} \tag{7}$$

$$\begin{gathered}\mathtt{t}:\mathtt{Super}-\mathtt{Peer}(?\mathtt{sp}) \wedge \mathtt{p}:\mathtt{hasOwnPeers}(?\mathtt{sp},?\mathtt{p}) \wedge \mathtt{t}:\mathtt{Peer}(?\mathtt{p}) \\ \wedge \mathtt{t}:\mathtt{canFind}(?\mathtt{p},?\mathtt{da}) \wedge \mathtt{t}:\mathtt{Data}(?\mathtt{da}) \wedge \mathtt{t}:\mathtt{hasPII}(?\mathtt{da},?\mathtt{pii}) \wedge \mathtt{t}:\mathtt{PII}(?\mathtt{pii}) \\ \longrightarrow \mathtt{p}:\mathtt{hasDisclosedFor}(?\mathtt{sp},?\mathtt{pii})\end{gathered} \tag{8}$$

$$\begin{gathered}\mathtt{t}:\mathtt{DataPolicy}(?\mathtt{dap}) \wedge \mathtt{t}:\mathtt{isBelongedTo}(?\mathtt{dap},?\mathtt{dmp}) \wedge \mathtt{t}:\mathtt{DomainPolicy}(?\mathtt{dmp}) \\ \wedge \mathtt{p}:\mathtt{domainPolicyForSPD}(?\mathtt{dmp},?\mathtt{spd}) \longrightarrow \mathtt{p}:\mathtt{dataPolicyForSPD}(?\mathtt{dap},?\mathtt{spd})\end{gathered} \tag{9}$$

$$\begin{gathered}\mathtt{t}:\mathtt{Request}(?\mathtt{r}) \wedge \mathtt{p}:\mathtt{getInTo}(?\mathtt{r},?\mathtt{spd}) \wedge \mathtt{t}:\mathtt{satisfy}(?\mathtt{r},?\mathtt{dap}) \wedge \mathtt{t}:\mathtt{DataPolicy}(?\mathtt{dap}) \\ \wedge \mathtt{p}:\mathtt{dataPolicyForSPD}(?\mathtt{dap},?\mathtt{spd}) \wedge \mathtt{t}:\mathtt{SPD}(?\mathtt{spd}) \wedge \mathtt{t}:\mathtt{hasSuperPeer}(?\mathtt{spd},?\mathtt{sp}) \\ \wedge \mathtt{p}:\mathtt{hasDisclosedFor}(?\mathtt{sp},?\mathtt{pii}) \longrightarrow \mathtt{p}:\mathtt{canUse}(?\mathtt{r},?\mathtt{pii})\end{gathered} \tag{10}$$

7.3. Policy Exceptions Handling

In formalizing access control policies, we may confront a situation in which a given request is neither explicitly allowed nor explicitly denied. A default decision must be made, as in the default *open* and *closed* policies, where authorization is respectively granted or denied by default.

The layers induced by Datalog stratification may be regarded as the steps of a methodology for constructing open policies in a principled way, starting with explicit authorizations, unless exception occurs, and adding derived authorizations through inheritance along hierarchies of subjects, objects, purposes, and rules. This approach can clearly implement defeasible inheritance, as shown in Section 7.4.

In general, the computational complexity of DL non-monotonic reasoning is very high, and the major DL reasoning engines do not support non-monotonic reasoning [5]; therefore, we apply stratified *Datalog*$^{\neg}$ to address defeasible inheritance when semantic legal policies are unified in the $sp_{\alpha\cap\beta}$ of $SPD_{\alpha\cap\beta}$.

7.4. Non-Monotonic Reasoning in $SPD_{\alpha\cap\beta}$

Once a Taiwan national security officer enters an $SPD_{\alpha\cap\beta}$, he/she must simultaneously comply with Singapore data protection laws α and Taiwan national security laws β. Here, we apply stratified *Datalog*$^{\neg}$ in Rule 6 of Section 7.2 for policy exceptions handling to comply with both of the above-mentioned two laws. In closed-world-assumption (CWA) semantics, the absence of consent is a weak negation ($\sim$), indicated as false, e.g., $\perp$.

We demonstrate how two exceptions (strata) are applied for possible dataset disclosure. In stratum one, according to the closed data protection policy, we do not disclose the personal dataset A to a data user u unless a data owner's explicit prior consent for a particular purpose p was obtained. In fact, this is the original principle of the data protection policy.

In Rule (6) of Section 7.2, an abnormal data request's condition $?\mathtt{c} = \mathtt{Ab1}$ in $\mathtt{t} : \mathtt{hasCondition}(?\mathtt{r}, ?\mathtt{c})$ and $\mathtt{t} : \mathtt{Condition}(?\mathtt{c})$ can be indicated as follows:

$$\begin{cases} hasPartOf.Condition(Ab1) \\ hasPartOf^{-}.Purpose(p) \\ hasPartOf^{-}.DataUser(u) \\ hasPartOf^{-}.Consent(\top) \end{cases}$$

In stratum two, we apply the default open national security policy to disclose the dataset C to Taiwan national security officers, even if we lack a data owner's explicit prior consent, *i.e.*, weak negation ($\sim$) indicated as $\bot$. However, we deny the Taiwan national security officer's request to disclose the dataset D for alien citizens, e.g., strong negation indicated as $\neg TW - citizenship$. Therefore, under Taiwanese national security laws, data will be legally disclosed unless a data request has its condition attributes satisfied by Ab2. Similarly, an abnormal data request's condition $?\mathtt{c} = \mathtt{Ab2}$ can be indicated as follows:

$$\begin{cases} hasPartOf.Condition(Ab2) \\ hasPartOf^{-}.Purpose(nationalSecurity) \\ hasPartOf^{-}.DataUser(securityOfficer) \\ hasPartOf^{-}.Consent(\bot) \\ hasPartOf^{-}.Nationality(\neg TW - citizenship) \end{cases}$$

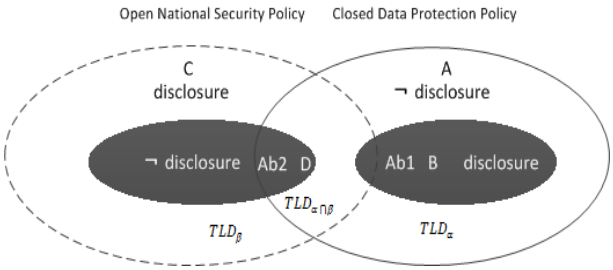
In Section 2.4, we show that Reiter-style default logic can be implemented in cq-programs to support non-monotonic reasoning of description logics. Furthermore, a prioritized default theory $\Delta = (\mathcal{T}, \mathcal{D}, \prec)$ resolves possible default reasoning conflicts from a finite set of prioritized defaults in $\mathcal{D}$. $\mathcal{T}$ is the DL-based ontology in the cq-program of $(\mathcal{T}, \mathcal{P})$ and $\mathcal{P}$ consists of a finite set of non-monotonic datalog rules (see Section 2.5).

In (11), DL-based $\mathcal{T}_\alpha$ ontology describes what is the concept of a disclosed PII_B set that satisfies a data protection policy (see Figure 9). $PII_{disclosure}$ and $PII_{\neg disclosure}$ are mutually exclusive.

$$\mathcal{T}_\alpha = \begin{cases} PII_{disclosure}(?pii) \sqsubseteq \neg PII_{\neg disclosure}(?pii) \\ PII_B(?pii) \sqsubseteq PII_A(?pii) \\ PII_B \sqsubseteq \neg PII_{\neg disclosure}(?pii) \\ PII_A(Alice) \end{cases} \tag{11}$$

$$\delta_0 = \frac{PII_A(?pii) : PII_{\neg disclosure}(?pii)}{PII_{\neg disclosure}(?pii)} \tag{12}$$

Figure 9. The final PII dataset disclosure is compliant with privacy protection and national security policies following by the priority ordering default reasoning.



On the one hand, an individual's *PII*, unless specified as an exception, is normally in a ¬*disclosure* PII_A set by a closed data protection policy's assumption in TLD_α [see default δ_0 in (12)]. P_{Ω_α} consists the following single rule for δ_0:

$$in_{PII_{\neg disclosure}}(?pii) \longleftarrow DL[\lambda;\ PII_A](?pii), not\ \ DL[\lambda;\ PII_{disclosure}](?pii) \tag{13}$$

In a nutshell, a cq-program provides two way information flow between ontologies and rules in the integrated knowledge base. A default δ_0 can be enforced as Rule (13), where $DL[\lambda;\ PII_A](?pii)$ is a *cq-atom* with input list of update predicate λ and PII_A is a *cq-query*. Auxiliary predicate $in_{PII_{\neg disclosure}}(?pii)$ is used in λ. $\lambda = PII_{\neg disclosure} \uplus in_{PII_{\neg disclosure}} \wedge PII_{\neg disclosure} \ \uplus\ in_{PII_{disclosure}}$ is the update list of form $PII_{\neg disclosure}$ in $\mathcal{T}_\alpha$, where $\uplus$ (resp., $\uplus$) increases $PII_{\neg disclosure}$ (resp., $PII_{disclosure}$). The answer set is $I_{\omega_\alpha} = \{in_{PII_{\neg disclosure(Alice)}}\}$

Whenever we successfully enforce a closed data protection policy, an individual's *PII* is included in a *disclosure* set, PII_B. Otherwise, it is still in a ¬*disclosure* set, PII_A. We add the following Rule (14) in P_{Ω_α} to achieve this objective:

$$\left\{ \begin{array}{l} in_{PII_{disclosure}}(?pii) \longleftarrow Request(?c, ?pii), Action_{ConditionCheck(Ab1,?pii)} \\ PII_B(?pii) \longleftarrow Action_{ConditionCheck(Ab1,?pii)}(\top) \\ Action_{ConditionCheck(Ab1,Alice)}(\bot), Action_{ConditionCheck(Ab1,Bob)}(\top) \end{array} \right\} \tag{14}$$

where $Action_{ConditionCheck(Ab1,?pii)}(\top)$ is to verify whether a request with its carrying context satisfies the `Ab1` by checking against some facts present at the beginning of the reasoning process in the knowledge base, which are fed by external mechanism. The default extension answer set is $I_{\omega_\alpha} = \{in_{PII_{\neg disclosure(Alice)}}, in_{PII_{disclosure(Bob)}}\}$.

In (15), DL-based $\mathcal{T}_\beta$ ontology describes what is the concept of a not disclosed PII_D set that satisfies a national security policy.

$$\mathcal{T}_\beta = \left\{ \begin{array}{l} PII_{disclosure}(?pii) \sqsubseteq \neg PII_{\neg disclosure}(?pii) \\ PII_D(?pii) \sqsubseteq PII_C(?pii) \\ PII_D \sqsubseteq PII_{\neg disclosure}(?pii) \\ PII_C(Alice) \end{array} \right\} \tag{15}$$

$$\delta_1 = \frac{PII_C(?pii) : PII_{disclosure}(?pii)}{PII_{disclosure}(?pii)} \tag{16}$$

On the other hand, an individual's *PII*, unless specified as an exception, is normally in a *disclosure* PII_C set by an open national security policy's assumption in TLD_β (see default δ_1 in 16). P_{Ω_β} consists the rule for δ_1:

$$in_{PII_{disclosure}}(?pii) \longleftarrow DL[\lambda;\ PII_C](?pii), not\ \ DL[\lambda;\ PII_{\neg disclosure}](?pii) \tag{17}$$

A default δ_1 can be enforced as Rule (17), where $DL[\lambda;\ \ PII_C](?pii)$ is a *cq-atom* with input list of update predicate λ and PII_C is a *cq-query*. Auxiliary predicate $in_{PII_{disclosure}}(?pii)$ is used in the input list of auxiliary predicate λ. $\lambda = PII_{disclosure} \uplus in_{PII_{disclosure}} \wedge PII_{disclosure} \ \uplus\ in_{PII_{\neg disclosure}}$ is the update lists of form $PII_{disclosure}$ in $\mathcal{T}_\beta$, where $\uplus$ (resp., $\uplus$) increases $PII_{disclosure}$ (resp., $PII_{\neg disclosure}$). The answer set $I_{\omega_\beta} = \{in_{PII_{disclosure(Alice)}}\}$.

Whenever we enforce an open national security policy with the satisfaction of Ab2, an individual's *PII* is included in a $\neg$*disclosure* set, PII_D. Otherwise, it is still in a *disclosure* set, PII_C. We add the following Rule (18) in P_{Ω_β} to achieve this objective:

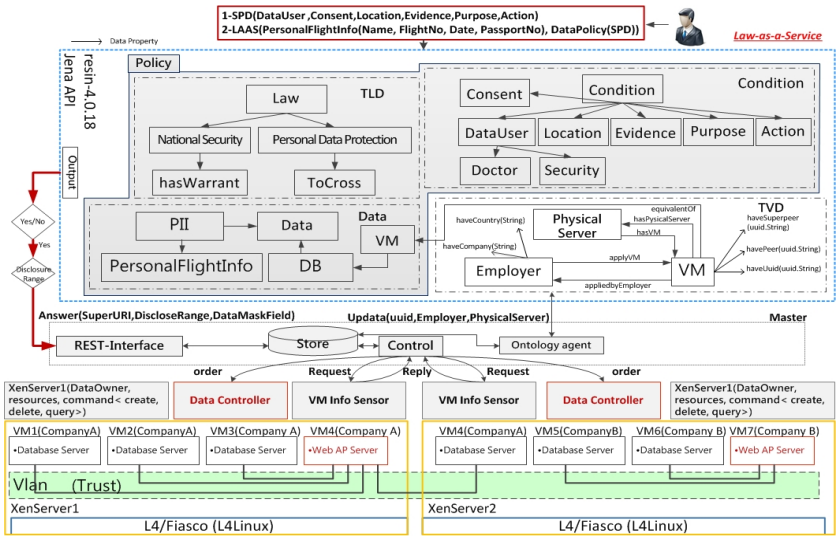
$$\left\{ \begin{array}{l} in_{PII_{\neg disclosure}}(?pii) \longleftarrow Request(?c, ?pii), Action_{ConditionCheck(Ab2,?pii)} \\ PII_D(?pii) \longleftarrow Action_{ConditionCheck(Ab2,?pii)}(\top) \\ Action_{ConditionCheck(Ab2,Alice)}(\bot), Action_{ConditionCheck(Ab2,David)}(\top) \end{array} \right\} \quad (18)$$

where $Action_{ConditionCheck(Ab2,?pii)}(\top)$ is to verify whether a request with its carrying context satisfies the Ab2 by checking against some facts present at the beginning of the reasoning process in the knowledge base, which are fed by external mechanism. The default extension answer set is $I_{\omega_\beta} = \{in_{PII_{disclosure(Alice)}}, in_{PII_{\neg disclosure(David)}}\}$. We have default extension conflict in $I_{\omega_\alpha} = \{in_{PII_{\neg disclosure(Alice)}}, \cdots\}$ and $I_{\omega_\beta} = \{in_{PII_{disclosure(Alice)}}, \cdots\}$. Strict priority ordering defaults resolve this *PII* disclosure conflict while enforcing different default extension logic reasoning. In this study, the priority order is compliant with the national security policy, which is prioritized over the data protection policy; thus, we have $\delta_1 \prec \delta_0$. Therefore, $PII_{\neg disclosure(Alice)} \in I_{\omega_\alpha}$ is false. It is impossible to have an individual's *PII* in a disclosure set and a $\neg$disclosure set after default extensions are complete.

7.5. LaaS Implementation

LaaS has been successfully implemented in semantic policy infrastructure to verify this concept (see Figure 10). Semantic legal policy enforcement is the mapping from a data usage context to access control decisions, including permit, deny, and error. A data usage context comprises a user's role along with his/her personal properties, resources metadata, access time, access location, purpose, and action. A data usage context is created when a user asks for information at the super-peer. A user's inputs for information queries constitute data usage context, *i.e.*, sets of ground facts (or instances) fed into the policies for outputs. The possible outputs from the semantic legal policy reasoning are sometimes more than simple answers like yes, no or unknown. They might provide explanations for query results.

Figure 10. Semantic legal policies are expressed as logical theories of cq-programs, e.g., OWL-DL ontologies and stratified Datalog rules with negation, for information queries.



8. Related Work

In a previous study [37], semantics-enabled enterprise cloud management fulfils the challenges of intelligent information management, especially regarding the issues of data integration, intelligent information access, and analytics. However, it does not consider enforcing law compliant semantic legal policies while providing automated resources self-managed services.

The Legal Knowledge Interchange Format (LKIF) uses a Semantic Web language to represent legal knowledge and thus support the modeling of legal domains [38]. The LKIF extends Semantic Web Rule Language (SWRL) [39] with support for negation and defeasible reasoning. In this study, OWL-DL policy ontologies are used as terminological knowledge for legal norm representations, and default logic in the cq-program is used for policy exceptions handling through non-monotonic reasoning [11,23].

In SemPIF [40], a meta-policy is a policy about policies that provides a set of rules for realizing services needed to manage policies. Moreover, a meta-policy consists of a set of rules for setting up the priorities of polices to be coordinated. Unlike Datalog rules, a meta-policy is only used for policy conflict resolutions and not for defeasible inheritance within ontologies and rules.

In another study [41], privacy policies are expressed as a first-order logic. Privacy expectation can be expressed using context information norms. An information flow satisfies privacy expectations if any one positive norm and all negative norms applicable to the transmission context are satisfied. Both positive and negative norms may also contain exceptions [42]. Here, we apply non-monotonic cq-programs for policy exceptions handling. In fact, default logic and CWA can be implemented in cq-programs to support non-monotonic reasoning for description logics.

9. Conclusions and Future Work

We extend our previous work [7] and provide legalized data exchange 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. A pandemic investigation scenario is demonstrated to explain why the LaaS is applicable for making a dataset disclosure decision either within a single jurisdiction or across jurisdictions.

Semantic Web technologies are applied to the semantic legal policy representation for data exchange and protection. The semantic legal policies are represented as a combination of ontologies and stratified Datalog rules with negation (or *Datalog*$^{\neg}$). More specifically, we use cq-programs with default logic reasoning over description logic for policy exceptions handling.

In the semantic cloud infrastructure, semantic legal policies are enforced in the super-peer to enable Law-as-a-Service (LaaS) and subsequent queries for CSPs and their customers. The agent at the law-aware super-peer is a unique guardian that provides data integration and protection services for its peers within a super-peer domain. Each agent at the super-peer also offers data exchange and protection services across super-peer domains.

Future work includes further exploiting the non-monotonic reasoning of policy exceptions handling and the expressive power of semantic legal policy under a hybrid integration of ontologies and non-monotonic rules.

Acknowledgements

This research was partially supported by the NSC Taiwan under Grant No. NSC 100-2221-E-004-011-MY2. A preliminary version of this paper appeared in the International Conference on Web Intelligence, Mining and Semantics (WIMS'12) [7] with the ACM DOI 10.1145/2254129.2254162 shown in the ACM digital library.

References

1. Eberhart, A.; Haase, P.; Oberle, D.; Zacharias, V. Semantic technologies and cloud computing. In *Foundations for the Web of Information and Services*; Fensel, D., Ed.; Springer: Berlin, Germany, 2011; pp. 239–251.
2. Abbadi, M.I. Self-managed services conceptual model in trustworthy clouds' infrastructure. In *Proceedings of Workshop on Cryptography and Security in Clouds*, Zurich, Switzerland, 15–16 March 2011.
3. Cabuk, S.; Dalton, C.I.; Eriksson, K.; Kuhlmann, D.; Ramasamy, H.V.; Ramunno, G.; Sadeghi, A.R.; Schunter, M.; Stüble, C. Towards automated security policy enforcement in multi-tenant virtual data centers. *J. Comput. Secur.* **2010**, *18*, 89–121.
4. Calvanese, D.; de Giacomo, D.; Lenzerini, M.; Rosati, R. View-based query answering over description logic ontologies. In *Proceedings of Eleventh International Conference on Principles of Knowledge Representation and Reasoning*, Sydney, Australia, 16–19 September 2008.
5. Bonatti, A.P. Datalog for security, privacy and trust. *Datalog Reloaded* **2011**, *6702*, 21–36.
6. Hu, Y.J.; Wu, W.N.; Yang, J.J. Semantics-enabled policies for information sharing and protection in the cloud. *Lect. Notes Comput. Sci.* **2011**, *6984*, 198–211.
7. Hu, Y.J.; Wu, W.N.; Cheng, D.R. Towards law-aware semantic cloud policies with exceptions for data integration and protection. In *Proceedings of International Conference on Web Intelligence, Mining and Semantics (WIMS12)*, Craiova, Romania, 13–15 June 2012.
8. Popp, R.; Poindexter, J. Countering terrorism through information and privacy protection technologies. *IEEE Secur. Priv.* **2006**, *4*, 24–33.
9. Peter Fleischer's Blog: Which Privacy Laws Should Apply on the Global Internet? Available online: http://peterfleischer.blogspot.com (accessed on 19 October 2012).
10. Pollock, L.J. Defeasible reasoning. In *Reasoning: Studies of Human Inference and Its Foundations*; Adler, J., Rips, L., Eds.; Cambridge University Press: New York, NY, USA, 2008.
11. Drabent, W.; Eiter, T.; Ianni, G.; Krennwallner, T.; Lukasiewicz, T.; Mauszyǹski, J. Hybrid reasoning with rules and ontologies. *Semant. Tech. Web* **2009**, *5500*, 1–49.
12. Calvanese, D.; de Giacomo, G.; Lembo, D.; Lenzerini, M.; Rosati, R. *Data Management in Peer-to-Peer Data Integration Systems*; IOS Press: Amsterdam, The Netherlands, 2006; pp. 177–201.
13. Halevy, A.; Ives, Z.G.; Madhavan, J.; Mork, P.; Suciu, D.; Tatarinov, I. The Piazza Peer data management system. *IEEE Trans. Knowled. Data Eng.* **2004**, *16*, 787–798.

14. Madhavan, J.; Jeffery, S.R.; Cohen, S.; Dong, X.; Ko, D.; Yu, C.; Halevy, A. Web-scale data integration: You can only afford to pay as you go. In *Proceedings of Third Biennial Conference on Innovative Data Systems Research*, Asilomar, CA, USA, 7–10 January 2007.
15. Halevy, Y.A. Answering queries using views: A survey. *VLDB J.* **2001**, *10*, 270–294.
16. Lenzerini, M. Data integration: A theoretical perspective. In *Proceedings of the ACM Symposium on Principles of Database Systems*, Madison, WI, USA, 3–5 June 2002.
17. Friedman, M.; Levy, A.; Millstein, T.; Navigational plans for data integration. In *Proceedings of the 16th National Conference on Artificial Intelligence*, Orlando, Fl ,USA, 19–22 July 1999.
18. Faigin, R.; Kolaitis, P.G.; Miller, R.J.; Popa, L. Data exchange: Semantics and query answering. *Theor. Comput. Sci.* **2005**, *336*, 89–124.
19. Clifton, C.; Kantarcioğlu, M.; Doan, A.; Schadow, G.; Vaidya, J.; Elmagarmid, A.; Suciu, D. Privacy-preserving data integration and sharing. In *Proceedings of 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, Paris, France, 13 June 2004.
20. Nash, A.; Deutsch, A. Privacy in GLAV Information integration. *Lect. Notes Comput. Sci.* **2006**, *4353*, 89–103.
21. Ceri, S.; Gottlob, G.; Tanca, L. What you always wanted to know about Datalog (and never dared to ask). *IEEE Trans. Knowl. Data Eng.* **1989**, *1* 146–166.
22. Meditskos, G.; Bassilliades, N. Rule-based OWL ontology reasoning systems: Implementations, strength, and weakness. In *Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches*; IGI Global: Hershey, PA, USA, 2009; pp. 124–148.
23. Dao-Tran, M.; Eiter, T.; Krennwallner, T. Realizing default logic over description logic knowledge bases. *Lect. Notes Comput. Sci.* **2009**, *5590*, 602–613.
24. Antoniou, G. *Nonmontonic Reasoning*; The MIT Press: Cambridge, MA, USA, 1997.
25. Brewka, G. Reasoning about priorities in default logic. In *Proceedings of 12th National Conference on Artificial Intelligence*, Seattle, WA, USA, 31 July–4 August 2012.
26. Weitzner, J.D.; Hendler, J. Creating a policy-aware web: Discretionary, rule-based access for the World Wide Web. In *Web and Information Security*; Ferrari, E., Thuraisingham, B., Eds.; IGI Global: Hershey, PA, USA, 2006; pp. 1–31.
27. Halevy, A.; Ives, Z.G.; Suciu, D.; Tatarinov, I. Schema mediation in peer data management systems. In *Proceedings of 19th International Conference on Data Engineering (ICDE)*, Bangalore, India, 5–8 March 2003; pp. 505–516.
28. Beneventano, D.; Bergamaschi, S.; Guerra, F.; Vincini, M. Querying a super-peer in a schema-based super-peer network. *Lect. Notes Comput. Sci.* **2007**, *4125*, 13–25.
29. Euzenat, J.; Shvaiko, P. *Ontology Matching*; Springer: Berlin, Germany, 2007.
30. Hu, Y.J.; Yang, J.J. A semantic privacy-preserving model for data sharing and integration. In *Proceedings of International Conference on Web Intelligence, Mining and Semantics*, Sogndal, Norway, 25–27 May 2011.
31. Foresti, S. *Preserving Privacy in Data Outsourcing*; Springer: Berlin, Germany, 2011.
32. Goasdoué, F.; Rousset, M.C. Answering queries using views: A KRDB perspective for the semantic web. *ACM Trans. on Internet Technol.* **2004**, *4*, 255–288.

33. Di Vimercati, S.C.; Foresti, S.; Jajodia, S.; Samarati, P. Access control policies and languages in open environments. *Adv. Inf. Secur.* **2007**, *33*, 21–58.
34. Perry, J.W. *Protecting Individual Privacy in the Struggle Against Terrorists: A Framework for Program Assessment*; The National Academies Press: Washington, DC, USA, 2008.
35. Deyrup, I.; Matthew, S. *Cloud Computing and National Security Laws*; Technical report; The Harvard Law National Security Research Group: Cambridge, MA, USA, 2010.
36. Sequeda, F.J.; Tirmizi, S.H.; Corcho, O.; Miranker, D.P. Survey of directly mapping SQL databases to the Semantic Web. *Knowl. Eng. Rev.* **2011**, *26*, 445–486.
37. Haase, P.; Mathäß, T.; Schmidt, M.; Eberhart, A.; Walther, U. Semantic technologies for enterprise cloud management. *Lect. Notes Comput. Sci.* **2010**, *6497*, 98–113.
38. Boer, A. *Legal Theory: Sources of Law and the Semantic Web*; IOS Press: Amsterdam, The Netherlands, 2009.
39. Horrocks, I.; Patel-Schneider, P.F.; Boley, H.; Tabet, S.; Grosof, B.; Dean, M. SWRL: A semantic web rule language combing OWL and RuleML. *World Wide Web* **2004**. Available online: http://www.w3.org/Submission/SWRL/ (accessed on 19 October 2012).
40. Hu, Y.J.; Boley, H. SemPIF: A semantic meta-policy interchange format for multiple web policies. In *Proceedings of Web Intelligence and Intelligent Agent Technology (WI-IAT)*, Toronto, Canada, 31 August–3 September 2010; pp. 302–307.
41. Barth, A.; Datta, A.; Mitchell, J.C.; Nissenbaum, H. Privacy and contextual integrity: Framework and applications. In *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 21–24 May 2006.
42. Datta, A.; Blocki, J.; Christin, N.; DeYoung, H.; Garg, D.; Jia, L.; Kaynar, D.; Sinha, A. Understanding and protecting privacy: Formal semantics and principled audit mechanisms. *Lect. Notes Comput. Sci.* **2011**, *7093*, 1–27.
43. Cali, A.; Gottlob, G.; Lukasiewicz, T.; Marnette, B.; Pieris, A. *Datalog*$^{+-}$: A family of logical knowledge representation and query languages for new applications: Keynote lecture. In *Proceedings of 25th annual IEEE Symposium on Logic in Computer Science*, Edinburgh, UK, 11–14 July 2010.
44. Gordon, F.T. *The Legal Knowledge Interchange Format (LKIF)*; Technical report, Deliverable D4.1.; The European project for Standardized Transparent Representations in order to Extend Legal Accessibility (ESTRELLA): Amsterdam, The Netherlands, 2008.