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

Yuh-Jong Hu Win-Nan Wu Di-Rong Cheng
{hu, d9905, 98753031}@cs.nccu.edu.tw

Emerging Network Technology(ENT) Lab.
Department of Computer Science
National Chengchi University, Taipei, Taiwan

June-19-2012

Legal Informatics, School of Law
Alma Mater Studiorum Universita di Bologna
C.I.R.S.F.I.D.

Motivations

1. Current cloud infrastructures do not provide enough automatically self-managed services.
2. In order to seek technology innovation on Software-as-a-service (SaaS), we apply semantic web technologies for cloud computing.
3. Automatically self-managed SaaS is not only for automatic allocation of cloud resources, but also for enforcing security and privacy policies.
4. *Law-as-a-Service (LaaS)* further enhances security and privacy policy representation and enforcement in the cloud.

Motivations

1. Current cloud infrastructures do not provide enough automatically self-managed services.
2. In order to seek technology innovation on Software-as-a-service (SaaS), we apply semantic web technologies for cloud computing.
3. Automatically self-managed SaaS is not only for automatic allocation of cloud resources, but also for enforcing security and privacy policies.
4. *Law-as-a-Service (LaaS)* further enhances security and privacy policy representation and enforcement in the cloud.

Motivations

1. Current cloud infrastructures do not provide enough automatically self-managed services.
2. In order to seek technology innovation on Software-as-a-service (SaaS), we apply semantic web technologies for cloud computing.
3. Automatically self-managed SaaS is not only for automatic allocation of cloud resources, but also for enforcing security and privacy policies.
4. *Law-as-a-Service (LaaS)* further enhances security and privacy policy representation and enforcement in the cloud.

Motivations

1. Current cloud infrastructures do not provide enough automatically self-managed services.
2. In order to seek technology innovation on Software-as-a-service (SaaS), we apply semantic web technologies for cloud computing.
3. Automatically self-managed SaaS is not only for automatic allocation of cloud resources, but also for enforcing security and privacy policies.
4. *Law-as-a-Service (LaaS)* further enhances security and privacy policy representation and enforcement in the cloud.

Research Goals

1. How to empower semantic technologies for cloud computing to provide law-aware semantics-enabled cloud policies?
2. How to accomplish data protection while enforcing data integration?
3. How to use semantic legal policies to interpret laws and ensure the legality of data sharing and protection across jurisdictions?
4. How to unify semantic policies and allow defeasible reasoning of a policy's exceptions handling?

Research Goals

1. How to empower semantic technologies for cloud computing to provide law-aware semantics-enabled cloud policies?
2. How to accomplish data protection while enforcing data integration?
3. How to use semantic legal policies to interpret laws and ensure the legality of data sharing and protection across jurisdictions?
4. How to unify semantic policies and allow defeasible reasoning of a policy's exceptions handling?

Research Goals

1. How to empower semantic technologies for cloud computing to provide law-aware semantics-enabled cloud policies?
2. How to accomplish data protection while enforcing data integration?
3. How to use semantic legal policies to interpret laws and ensure the legality of data sharing and protection across jurisdictions?
4. How to unify semantic policies and allow defeasible reasoning of a policy's exceptions handling?

Research Goals

1. How to empower semantic technologies for cloud computing to provide law-aware semantics-enabled cloud policies?
2. How to accomplish data protection while enforcing data integration?
3. How to use semantic legal policies to interpret laws and ensure the legality of data sharing and protection across jurisdictions?
4. How to unify semantic policies and allow defeasible reasoning of a policy's exceptions handling?

Contributions

1. A law-aware semantic cloud policy infrastructure has been established to verify the feasibility of LaaS concepts.
2. Semantic legal policies for data integration and protection are designed and enforced in a super-peer architecture.
3. Constructing multiple super-peer domains to verify semantic legal policies across jurisdictions.
4. Exploiting stratified Datalog with negation for a policy's exceptions handling.

Contributions

1. A law-aware semantic cloud policy infrastructure has been established to verify the feasibility of LaaS concepts.
2. Semantic legal policies for data integration and protection are designed and enforced in a super-peer architecture.
3. Constructing multiple super-peer domains to verify semantic legal policies across jurisdictions.
4. Exploiting stratified Datalog with negation for a policy's exceptions handling.

Contributions

1. A law-aware semantic cloud policy infrastructure has been established to verify the feasibility of LaaS concepts.
2. Semantic legal policies for data integration and protection are designed and enforced in a super-peer architecture.
3. Constructing multiple super-peer domains to verify semantic legal policies across jurisdictions.
4. Exploiting stratified Datalog with negation for a policy's exceptions handling.

Contributions

1. A law-aware semantic cloud policy infrastructure has been established to verify the feasibility of LaaS concepts.
2. Semantic legal policies for data integration and protection are designed and enforced in a super-peer architecture.
3. Constructing multiple super-peer domains to verify semantic legal policies across jurisdictions.
4. Exploiting stratified Datalog with negation for a policy's exceptions handling.

A Law-Aware Semantic Policy Infrastructure

We proposed a three-layer law-aware semantic policy infrastructure in [25]:

- Trusted Legal Domain (TLD)
- Trusted Virtual Domain (TVD)
- Trusted Machine Domain (TMD).

A Law-Aware Semantic Policy Infrastructure

We proposed a three-layer law-aware semantic policy infrastructure in [25]:

- Trusted Legal Domain (TLD)
- Trusted Virtual Domain (TVD)
- Trusted Machine Domain (TMD).

A Law-Aware Semantic Policy Infrastructure

We proposed a three-layer law-aware semantic policy infrastructure in [25]:

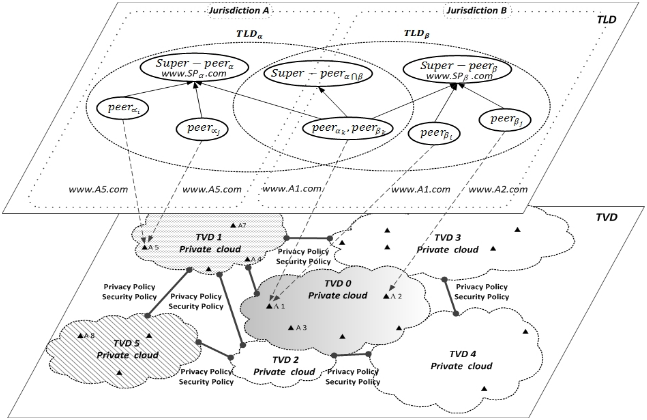
- Trusted Legal Domain (TLD)
- Trusted Virtual Domain (TVD)
- Trusted Machine Domain (TMD).

A Law-Aware Semantic Policy Infrastructure

We proposed a three-layer law-aware semantic policy infrastructure in [25]:

- Trusted Legal Domain (TLD)
- Trusted Virtual Domain (TVD)
- Trusted Machine Domain (TMD).

A Law-Aware Semantic Policy Infrastructure (conti.)



Logical Cage Model vs. Legal Cage Model

- A TVD is a *logical cage* model, which consists of a set of distributed virtual machines (VMs), storage for the VMs, and a communication medium interconnecting the VMs [6].
- A TLD is a *legal cage* model, which determined by a specific law, to regulate virtual legal boundary of data disclosure and usage.
- TLD concepts are modeled as a taxonomy of laws, where a type of law and an effective judicial domain are two factors to decide whether a data request is allowed.

Logical Cage Model vs. Legal Cage Model

- A TVD is a *logical cage* model, which consists of a set of distributed virtual machines (VMs), storage for the VMs, and a communication medium interconnecting the VMs [6].
- A TLD is a *legal cage* model, which determined by a specific law, to regulate virtual legal boundary of data disclosure and usage.
- TLD concepts are modeled as a taxonomy of laws, where a type of law and an effective judicial domain are two factors to decide whether a data request is allowed.

Logical Cage Model vs. Legal Cage Model

- A TVD is a *logical cage* model, which consists of a set of distributed virtual machines (VMs), storage for the VMs, and a communication medium interconnecting the VMs [6].
- A TLD is a *legal cage* model, which determined by a specific law, to regulate virtual legal boundary of data disclosure and usage.
- TLD concepts are modeled as a taxonomy of laws, where a type of law and an effective judicial domain are two factors to decide whether a data request is allowed.

Semantic Legal Policies as Logical Theories [5]

- Semantic legal policies are expressed as logical theories for information queries, and context are sets of ground facts that fed into policies for outputs.
- Semantic legal policies are mapping from a data usage context to access control decisions, such as 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, etc.
- 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.
- Semantic legal policy outputs (or query answers) are also encoded as logical formulas for authorization.

Semantic Legal Policies as Logical Theories [5]

- Semantic legal policies are expressed as logical theories for information queries, and context are sets of ground facts that fed into policies for outputs.
- Semantic legal policies are mapping from a data usage context to access control decisions, such as 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, etc.
- 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.
- Semantic legal policy outputs (or query answers) are also encoded as logical formulas for authorization.

Semantic Legal Policies as Logical Theories [5]

- Semantic legal policies are expressed as logical theories for information queries, and context are sets of ground facts that fed into policies for outputs.
- Semantic legal policies are mapping from a data usage context to access control decisions, such as 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, etc.
- 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.
- Semantic legal policy outputs (or query answers) are also encoded as logical formulas for authorization.

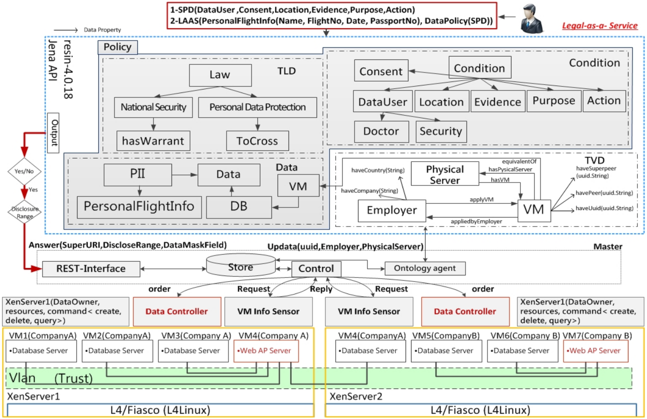
Semantic Legal Policies as Logical Theories [5]

- Semantic legal policies are expressed as logical theories for information queries, and context are sets of ground facts that fed into policies for outputs.
- Semantic legal policies are mapping from a data usage context to access control decisions, such as 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, etc.
- 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.
- Semantic legal policy outputs (or query answers) are also encoded as logical formulas for authorization.

Semantic Legal Policies as Logical Theories [5]

- Semantic legal policies are expressed as logical theories for information queries, and context are sets of ground facts that fed into policies for outputs.
- Semantic legal policies are mapping from a data usage context to access control decisions, such as 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, etc.
- 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.
- Semantic legal policy outputs (or query answers) are also encoded as logical formulas for authorization.

Semantic Legal Policies as Logical Theories (conti.)



Related Work

Several categories are related to this study:

- Multi-tenant cloud services: Abbadi [1], Cabuk [6], Eberhart [13], Foresti [16], Haase [20], Hu [25].
- Peer data management: Beneventano [3], Calvanese [7], Halevey [21] [22], Hu [27], Madhavan [31].
- Semantic policies for data sharing and protection: Clifton [10], Hu [24] [26].
- Semantic privacy policies: Bart [2], Datta [11], Weitzner [37].
- Semantic legal informatics: Boer [4], Gordon [19].
- Datalog for security and privacy: Bonatti [5], Jajodia [28].

Related Work

Several categories are related to this study:

- Multi-tenant cloud services: Abbadi [1], Cabuk [6], Eberhart [13], Foresti [16], Haase [20], Hu [25].
- Peer data management: Beneventano [3], Calvanese [7], Halevey [21] [22], Hu [27], Madhavan [31].
- Semantic policies for data sharing and protection: Clifton [10], Hu [24] [26].
- Semantic privacy policies: Bart [2], Datta [11], Weitzner [37].
- Semantic legal informatics: Boer [4], Gordon [19].
- Datalog for security and privacy: Bonatti [5], Jajodia [28].

Related Work

Several categories are related to this study:

- Multi-tenant cloud services: Abbadi [1], Cabuk [6], Eberhart [13], Foresti [16], Haase [20], Hu [25].
- Peer data management: Beneventano [3], Calvanese [7], Halevey [21] [22], Hu [27], Madhavan [31].
- Semantic policies for data sharing and protection: Clifton [10], Hu [24] [26].
- Semantic privacy policies: Bart [2], Datta [11], Weitzner [37].
- Semantic legal informatics: Boer [4], Gordon [19].
- Datalog for security and privacy: Bonatti [5], Jajodia [28].

Related Work

Several categories are related to this study:

- Multi-tenant cloud services: Abbadi [1], Cabuk [6], Eberhart [13], Foresti [16], Haase [20], Hu [25].
- Peer data management: Beneventano [3], Calvanese [7], Halevey [21] [22], Hu [27], Madhavan [31].
- Semantic policies for data sharing and protection: Clifton [10], Hu [24] [26].
- Semantic privacy policies: Bart [2], Datta [11], Weitzner [37].
- Semantic legal informatics: Boer [4], Gordon [19].
- Datalog for security and privacy: Bonatti [5], Jajodia [28].

Related Work

Several categories are related to this study:

- Multi-tenant cloud services: Abbadi [1], Cabuk [6], Eberhart [13], Foresti [16], Haase [20], Hu [25].
- Peer data management: Beneventano [3], Calvanese [7], Halevey [21] [22], Hu [27], Madhavan [31].
- Semantic policies for data sharing and protection: Clifton [10], Hu [24] [26].
- Semantic privacy policies: Bart [2], Datta [11], Weitzner [37].
- Semantic legal informatics: Boer [4], Gordon [19].
- Datalog for security and privacy: Bonatti [5], Jajodia [28].

Related Work

Several categories are related to this study:

- Multi-tenant cloud services: Abbadi [1], Cabuk [6], Eberhart [13], Foresti [16], Haase [20], Hu [25].
- Peer data management: Beneventano [3], Calvanese [7], Halevey [21] [22], Hu [27], Madhavan [31].
- Semantic policies for data sharing and protection: Clifton [10], Hu [24] [26].
- Semantic privacy policies: Bart [2], Datta [11], Weitzner [37].
- Semantic legal informatics: Boer [4], Gordon [19].
- Datalog for security and privacy: Bonatti [5], Jajodia [28].

A Super-Peer Domain (SPD) Model

A super-peer specifies its legal semantic policies based on a type of law from a jurisdiction within a super-peer domain:

- A Peer Data Management System (PDMS) is the best way to achieve wide-scale data integration over the Internet.
- However, a pure peer data integration architecture is hard to enact in the cloud environment because we are unable to capture the unstructured peer relationships from a large amount of peers.

A Super-Peer Domain (SPD) Model

A super-peer specifies its legal semantic policies based on a type of law from a jurisdiction within a super-peer domain:

- A Peer Data Management System (PDMS) is the best way to achieve wide-scale data integration over the Internet.
- However, a pure peer data integration architecture is hard to enact in the cloud environment because we are unable to capture the unstructured peer relationships from a large amount of peers.

A Super-Peer Domain (SPD) Model

A super-peer specifies its legal semantic policies based on a type of law from a jurisdiction within a super-peer domain:

- A Peer Data Management System (PDMS) is the best way to achieve wide-scale data integration over the Internet.
- However, a pure peer data integration architecture is hard to enact in the cloud environment because we are unable to capture the unstructured peer relationships from a large amount of peers.

Semantic Mappings from Local Schemas to Global Schema

Possible semantic mappings from local schemas to global schema:

- Global-As-View(GAV): expressing each concept in the global schema as queries over the data sources.
- Local-As-View(LAV): expressing each concept in the data sources as a query (or view) over the global schema.
- Global-Local-As-View(GLAV): allowing flexible schema definitions independent of the particular details of the data sources.

Semantic Mappings from Local Schemas to Global Schema

Possible semantic mappings from local schemas to global schema:

- Global-As-View(GAV): expressing each concept in the global schema as queries over the data sources.
- Local-As-View(LAV): expressing each concept in the data sources as a query (or view) over the global schema.
- Global-Local-As-View(GLAV): allowing flexible schema definitions independent of the particular details of the data sources.

Semantic Mappings from Local Schemas to Global Schema

Possible semantic mappings from local schemas to global schema:

- Global-As-View(GAV): expressing each concept in the global schema as queries over the data sources.
- Local-As-View(LAV): expressing each concept in the data sources as a query (or view) over the global schema.
- Global-Local-As-View(GLAV): allowing flexible schema definitions independent of the particular details of the data sources.

Semantic Mappings from Local Schemas to Global Schema

Possible semantic mappings from local schemas to global schema:

- Global-As-View(GAV): expressing each concept in the global schema as queries over the data sources.
- Local-As-View(LAV): expressing each concept in the data sources as a query (or view) over the global schema.
- Global-Local-As-View(GLAV): allowing flexible schema definitions independent of the particular details of the data sources.

Principles of Data Protection Laws

Three principles of data protection laws for cloud computing:

- *Registration* principle: location of service provider registration, which enables data collection services.
- *Nationality* principle: nationality of the data owner whose data are being used.
- *Territoriality* principle: data center location where actual data processing happens.

Principles of Data Protection Laws

Three principles of data protection laws for cloud computing:

- *Registration* principle: location of service provider registration, which enables data collection services.
- *Nationality* principle: nationality of the data owner whose data are being used.
- *Territoriality* principle: data center location where actual data processing happens.

Principles of Data Protection Laws

Three principles of data protection laws for cloud computing:

- *Registration* principle: location of service provider registration, which enables data collection services.
- *Nationality* principle: nationality of the data owner whose data are being used.
- *Territoriality* principle: data center location where actual data processing happens.

Principles of Data Protection Laws

Three principles of data protection laws for cloud computing:

- *Registration* principle: location of service provider registration, which enables data collection services.
- *Nationality* principle: nationality of the data owner whose data are being used.
- *Territoriality* principle: data center location where actual data processing happens.

Objectives of Law-Aware Semantic Cloud

Applying semantic technologies in the trusted virtual cloud infrastructure to:

1. offer LaaS for Cloud Service Providers (CSPs) while integrating semantic data modeled as ontologies from multiple data sources.
2. enable query services for cloud end-users through a combination of ontologies and stratified Datalog rules with negation.

Objectives of Law-Aware Semantic Cloud

Applying semantic technologies in the trusted virtual cloud infrastructure to:

1. offer LaaS for Cloud Service Providers (CSPs) while integrating semantic data modeled as ontologies from multiple data sources.
2. enable query services for cloud end-users through a combination of ontologies and stratified Datalog rules with negation.

Objectives of Law-Aware Semantic Cloud

Applying semantic technologies in the trusted virtual cloud infrastructure to:

1. offer LaaS for Cloud Service Providers (CSPs) while integrating semantic data modeled as ontologies from multiple data sources.
2. enable query services for cloud end-users through a combination of ontologies and stratified Datalog rules with negation.

A Super-Peer Data Cloud System

A super-peer data cloud system is a set of super-peer domains $\Pi = \{\pi_1, ..., \pi_n\}$, where

- Each super-peer domain π_i corresponds to a TLD.
- Grouping a set of peers into a super-peer domain and organize them into a two-level architecture: peers and super-peer.
- The super-peer is a guardian, which integrates all of its local peers' ontologies into a global ontology through ontology mapping, alignment, and merging.
- Semantic global mappings are also possible from the current $Super - peer_\alpha$ to interlink with another $Super - peer_\beta$.
- Semantic legal privacy policies enforcement is posed to a super-peer that provides data integration and protection services.

A Super-Peer Data Cloud System

A super-peer data cloud system is a set of super-peer domains $\Pi = \{\pi_1, ..., \pi_n\}$, where

- Each super-peer domain π_i corresponds to a TLD.
- Grouping a set of peers into a super-peer domain and organize them into a two-level architecture: peers and super-peer.
- The super-peer is a guardian, which integrates all of its local peers' ontologies into a global ontology through ontology mapping, alignment, and merging.
- Semantic global mappings are also possible from the current $Super - peer_\alpha$ to interlink with another $Super - peer_\beta$.
- Semantic legal privacy policies enforcement is posed to a super-peer that provides data integration and protection services.

A Super-Peer Data Cloud System

A super-peer data cloud system is a set of super-peer domains $\Pi = \{\pi_1, ..., \pi_n\}$, where

- Each super-peer domain π_i corresponds to a TLD.
- Grouping a set of peers into a super-peer domain and organize them into a two-level architecture: peers and super-peer.
- The super-peer is a guardian, which integrates all of its local peers' ontologies into a global ontology through ontology mapping, alignment, and merging.
- Semantic global mappings are also possible from the current $Super - peer_\alpha$ to interlink with another $Super - peer_\beta$.
- Semantic legal privacy policies enforcement is posed to a super-peer that provides data integration and protection services.

A Super-Peer Data Cloud System

A super-peer data cloud system is a set of super-peer domains $\Pi = \{\pi_1, ..., \pi_n\}$, where

- Each super-peer domain π_i corresponds to a TLD.
- Grouping a set of peers into a super-peer domain and organize them into a two-level architecture: peers and super-peer.
- The super-peer is a guardian, which integrates all of its local peers' ontologies into a global ontology through ontology mapping, alignment, and merging.
- Semantic global mappings are also possible from the current $Super - peer_\alpha$ to interlink with another $Super - peer_\beta$.
- Semantic legal privacy policies enforcement is posed to a super-peer that provides data integration and protection services.

A Super-Peer Data Cloud System

A super-peer data cloud system is a set of super-peer domains $\Pi = \{\pi_1, ..., \pi_n\}$, where

- Each super-peer domain π_i corresponds to a TLD.
- Grouping a set of peers into a super-peer domain and organize them into a two-level architecture: peers and super-peer.
- The super-peer is a guardian, which integrates all of its local peers' ontologies into a global ontology through ontology mapping, alignment, and merging.
- Semantic global mappings are also possible from the current $Super - peer_\alpha$ to interlink with another $Super - peer_\beta$.
- Semantic legal privacy policies enforcement is posed to a super-peer that provides data integration and protection services.

A Super-Peer Data Cloud System

A super-peer data cloud system is a set of super-peer domains $\Pi = \{\pi_1, ..., \pi_n\}$, where

- Each super-peer domain π_i corresponds to a TLD.
- Grouping a set of peers into a super-peer domain and organize them into a two-level architecture: peers and super-peer.
- The super-peer is a guardian, which integrates all of its local peers' ontologies into a global ontology through ontology mapping, alignment, and merging.
- Semantic global mappings are also possible from the current $Super - peer_\alpha$ to interlink with another $Super - peer_\beta$.
- Semantic legal privacy policies enforcement is posed to a super-peer that provides data integration and protection services.

Semantics of a TLD

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 semantic legal policies.
- Through local LAV mapping assertions, a global schema GS_α provides an integrated view for a set of peers from $P_\alpha = \{peer_1, ..., peer_n\}$.
- 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 $DS_\alpha = \{ds_1, ..., ds_m\}$.
- 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.
- A set of local data sources, ds_i, from DS_α, are relational structure data that store materialized data instances.

Semantics of a TLD

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 semantic legal policies.
- Through local LAV mapping assertions, a global schema GS_α provides an integrated view for a set of peers from $P_\alpha = \{peer_1, ..., peer_n\}$.
- 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 $DS_\alpha = \{ds_1, ..., ds_m\}$.
- 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.
- A set of local data sources, ds_i, from DS_α, are relational structure data that store materialized data instances.

Semantics of a TLD

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 semantic legal policies.
- Through local LAV mapping assertions, a global schema GS_α provides an integrated view for a set of peers from $P_\alpha = \{peer_1, ..., peer_n\}$.
- 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 $DS_\alpha = \{ds_1, ..., ds_m\}$.
- 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.
- A set of local data sources, ds_i, from DS_α, are relational structure data that store materialized data instances.

Semantics of a TLD

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 semantic legal policies.
- Through local LAV mapping assertions, a global schema GS_α provides an integrated view for a set of peers from $P_\alpha = \{peer_1, ..., peer_n\}$.
- 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 $DS_\alpha = \{ds_1, ..., ds_m\}$.
- 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.
- A set of local data sources, ds_i, from DS_α, are relational structure data that store materialized data instances.

Semantics of a TLD

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 semantic legal policies.
- Through local LAV mapping assertions, a global schema GS_α provides an integrated view for a set of peers from $P_\alpha = \{peer_1, ..., peer_n\}$.
- 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 $DS_\alpha = \{ds_1, ..., ds_m\}$.
- 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.
- A set of local data sources, ds_i, from DS_α, are relational structure data that store materialized data instances.

Semantics of a TLD

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 semantic legal policies.
- Through local LAV mapping assertions, a global schema GS_α provides an integrated view for a set of peers from $P_\alpha = \{peer_1, ..., peer_n\}$.
- 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 $DS_\alpha = \{ds_1, ..., ds_m\}$.
- 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.
- A set of local data sources, ds_i, from DS_α, are relational structure data that store materialized data instances.

Semantics of Multiple TLDs

A super-peer domain π_α for TLD_α is related to another super-peer domain π_β for TLD_β through:

- A set of super-peer's GLAV semantic mapping assertions

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

  ,
  where $CQ_{\pi_\beta}(sp_\beta)$ and $CQ_{\pi_\alpha}(sp_\alpha)$ are conjunctive queries over the super-peer sp_β and super-peer sp_α.
- A Datalog rule is a mapping assertion of GLAV:

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

  ,
  where H, query results (or views) are from the source of sp_α's global ontology schema, and rule antecedent B_i, is a pattern matching specification from target sp_β's global ontology schema.

Semantics of Multiple TLDs

A super-peer domain π_α for TLD_α is related to another super-peer domain π_β for TLD_β through:

- A set of super-peer's GLAV semantic mapping assertions

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

  ,
  where $CQ_{\pi_\beta}(sp_\beta)$ and $CQ_{\pi_\alpha}(sp_\alpha)$ are conjunctive queries over the super-peer sp_β and super-peer sp_α.

- A Datalog rule is a mapping assertion of GLAV:

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

  ,
  where H, query results (or views) are from the source of sp_α's global ontology schema, and rule antecedent B_i, is a pattern matching specification from target sp_β's global ontology schema.

Semantics of Multiple TLDs

A super-peer domain π_α for TLD_α is related to another super-peer domain π_β for TLD_β through:

- A set of super-peer's GLAV semantic mapping assertions

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

  ,
  where $CQ_{\pi_\beta}(sp_\beta)$ and $CQ_{\pi_\alpha}(sp_\alpha)$ are conjunctive queries over the super-peer sp_β and super-peer sp_α.

- A Datalog rule is a mapping assertion of GLAV:

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

  ,
  where H, query results (or views) are from the source of sp_α's global ontology schema, and rule antecedent B_i, is a pattern matching specification from target sp_β's global ontology schema.

Semantic Legal Policy Representation

1. A semantic legal policy is created from a policy language, and a semantic legal policy language is shown as a combination of ontology language and rule language.
2. A semantic legal policy is composed of ontologies and rules, where ontologies are created from an ontology language and rules are created from a rule language.
3. Currently, OWL-DL is used for policy ontology and stratified Datalog with negation, e.g., $Datalog^{\neg}$, rules are used for defeasible rules reasoning.
4. The research challenging is how to integrate two families of logics, description logic (DL) and logic program (LP), for a semantic legal policy representation and enforcement under non-monotonic semantics.

Semantic Legal Policy Representation

1. A semantic legal policy is created from a policy language, and a semantic legal policy language is shown as a combination of ontology language and rule language.
2. A semantic legal policy is composed of ontologies and rules, where ontologies are created from an ontology language and rules are created from a rule language.
3. Currently, OWL-DL is used for policy ontology and stratified Datalog with negation, e.g., $Datalog^{\neg}$, rules are used for defeasible rules reasoning.
4. The research challenging is how to integrate two families of logics, description logic (DL) and logic program (LP), for a semantic legal policy representation and enforcement under non-monotonic semantics.

Semantic Legal Policy Representation

1. A semantic legal policy is created from a policy language, and a semantic legal policy language is shown as a combination of ontology language and rule language.
2. A semantic legal policy is composed of ontologies and rules, where ontologies are created from an ontology language and rules are created from a rule language.
3. Currently, OWL-DL is used for policy ontology and stratified Datalog with negation, e.g., $Datalog^{\neg}$, rules are used for defeasible rules reasoning.
4. The research challenging is how to integrate two families of logics, description logic (DL) and logic program (LP), for a semantic legal policy representation and enforcement under non-monotonic semantics.

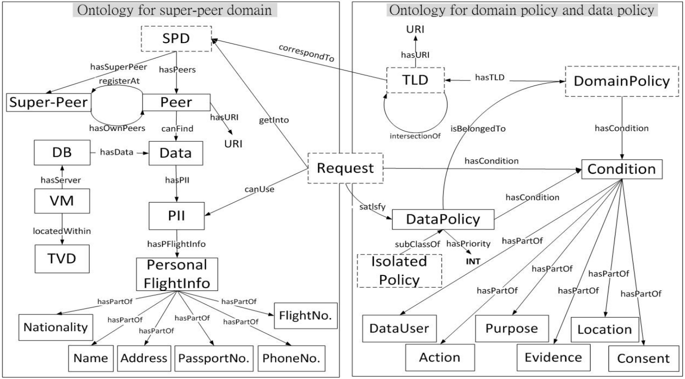
Semantic Legal Policy Representation

1. A semantic legal policy is created from a policy language, and a semantic legal policy language is shown as a combination of ontology language and rule language.
2. A semantic legal policy is composed of ontologies and rules, where ontologies are created from an ontology language and rules are created from a rule language.
3. Currently, OWL-DL is used for policy ontology and stratified Datalog with negation, e.g., $Datalog^{\neg}$, rules are used for defeasible rules reasoning.
4. The research challenging is how to integrate two families of logics, description logic (DL) and logic program (LP), for a semantic legal policy representation and enforcement under non-monotonic semantics.

Policy Ontology for a Super-Peer Domain

Semantics of a super-peer data cloud includes two modular concepts:

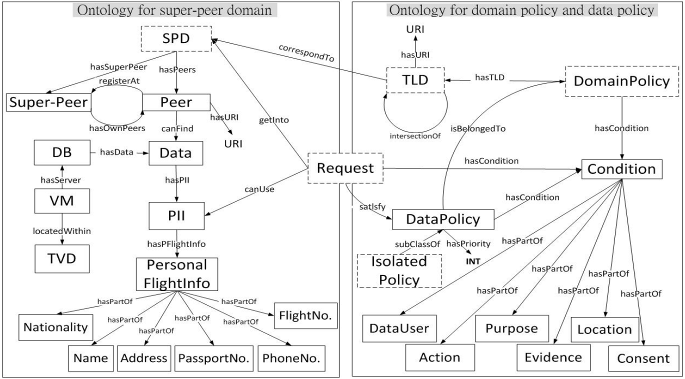
1. super-peer domain
2. domain policy and data policy



Policy Ontology for a Super-Peer Domain

Semantics of a super-peer data cloud includes two modular concepts:

1. super-peer domain
2. domain policy and data policy



Semantic Legal Policy Enforcement

- Balancing policy expressive power and computational complexity from integration of ontologies and rules.
- OWL-DL with positive unary and binary datalog rule from SWRL is not capable for a policy's exceptions handling.
- How about using different species of DL-Lite, e.g. $DL - Lite_A$, $DL - Lite_F$, $DL - Lite_R$ integrated with extended Datalog, $Datalog^{+-}$, for a semantic legal policy enforcement?
- Consider seriously about policy enforcement criteria in terms of computational complexity, such as undecidable vs. decidable, intractable vs. tractable, etc.

Semantic Legal Policy Enforcement

- Balancing policy expressive power and computational complexity from integration of ontologies and rules.
- OWL-DL with positive unary and binary datalog rule from SWRL is not capable for a policy's exceptions handling.
- How about using different species of DL-Lite, e.g. $DL - Lite_A$, $DL - Lite_F$, $DL - Lite_R$ integrated with extended Datalog, $Datalog^{+-}$, for a semantic legal policy enforcement?
- Consider seriously about policy enforcement criteria in terms of computational complexity, such as undecidable vs. decidable, intractable vs. tractable, etc.

Semantic Legal Policy Enforcement

- Balancing policy expressive power and computational complexity from integration of ontologies and rules.
- OWL-DL with positive unary and binary datalog rule from SWRL is not capable for a policy's exceptions handling.
- How about using different species of DL-Lite, e.g. $DL - Lite_A$, $DL - Lite_F$, $DL - Lite_R$ integrated with extended Datalog, $Datalog^{+-}$, for a semantic legal policy enforcement?
- Consider seriously about policy enforcement criteria in terms of computational complexity, such as undecidable vs. decidable, intractable vs. tractable, etc.

Semantic Legal Policy Enforcement

- Balancing policy expressive power and computational complexity from integration of ontologies and rules.
- OWL-DL with positive unary and binary datalog rule from SWRL is not capable for a policy's exceptions handling.
- How about using different species of DL-Lite, e.g. $DL - Lite_A$, $DL - Lite_F$, $DL - Lite_R$ integrated with extended Datalog, $Datalog^{+-}$, for a semantic legal policy enforcement?
- Consider seriously about policy enforcement criteria in terms of computational complexity, such as undecidable vs. decidable, intractable vs. tractable, etc.

Semantic Legal Policies

A Domain Policy's Ontology

A partial ontology for a domain policy

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

Semantic Legal Policies (conti.)

A Domain Policy's Rules (conti.)

Link between TLD and SPD

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

Request for an SPD

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

Semantic Legal Policies (conti.)

A Domain Policy's Rules (conti.)

Link between TLD and SPD

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

Request for an SPD

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

Semantic Legal Policies

A Data Policy's Ontology (conti.)

A partial ontology for a data policy

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

Semantic Legal Policies (conti.)

A Data Policy's Rules (conti.)

Super-peer has its own peers

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

Super-peer is allowed to disclose PII

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

Semantic Legal Policies (conti.)

A Data Policy's Rules (conti.)

Super-peer has its own peers

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

Super-peer is allowed to disclose PII

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

Semantic Legal Policies (conti.)

A Data Policy's Rules (conti.)

A Data policy for an SPD

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

Request can use PII

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

Semantic Legal Policies (conti.)

A Data Policy's Rules (conti.)

A Data policy for an SPD

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

Request can use PII

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

Unifying Two Types of Policies
Privacy Protection and National Security

1. We manually unify two types of semantic legal policies, translated from privacy protection law and national security law.
2. Privacy protection law α and national security law β are unified at $Super-peer_{\alpha\cap\beta}$ at $TLD_{\alpha\cap\beta}$, where $TLD_{\alpha\cap\beta}$ is in the intersection of TLD_{α} and TLD_{β} jurisdiction
3. Database is in compliance with a data protection law α from one jurisdiction but data centers hosting database are possibly in compliance with national security law β from another jurisdiction.

Unifying Two Types of Policies
Privacy Protection and National Security

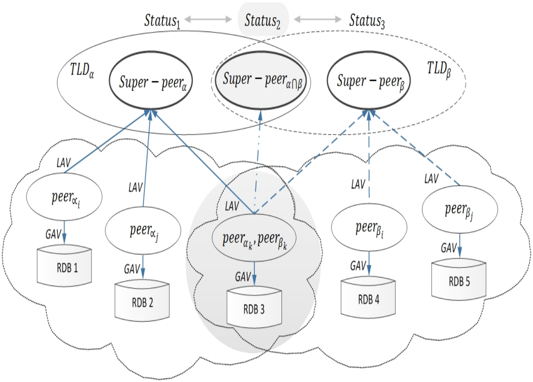
1. We manually unify two types of semantic legal policies, translated from privacy protection law and national security law.
2. Privacy protection law α and national security law β are unified at $\mathit{Super}-\mathit{peer}_{\alpha\cap\beta}$ at $TLD_{\alpha\cap\beta}$, where $TLD_{\alpha\cap\beta}$ is in the intersection of TLD_{α} and TLD_{β} jurisdiction
3. Database is in compliance with a data protection law α from one jurisdiction but data centers hosting database are possibly in compliance with national security law β from another jurisdiction.

Unifying Two Types of Policies

Privacy Protection and National Security

1. We manually unify two types of semantic legal policies, translated from privacy protection law and national security law.
2. Privacy protection law α and national security law β are unified at $Super - peer_{\alpha \cap \beta}$ at $TLD_{\alpha \cap \beta}$, where $TLD_{\alpha \cap \beta}$ is in the intersection of TLD_{α} and TLD_{β} jurisdiction
3. Database is in compliance with a data protection law α from one jurisdiction but data centers hosting database are possibly in compliance with national security law β from another jurisdiction.

Unifying Semantic Legal Policies at $Super - peer_{\alpha\cap\beta}$



Query at Intersection of TLDs

Two types of queries are available: subject-based and pattern-based:

1. At $Super - peer_{\alpha \cap \beta}$, only provides pattern-based queries, at $Super - peer_{\alpha}$ and $Super - peer_{\beta}$ we provide both.
2. A guardian agent in $Super - peer_{\alpha \cap \beta}$ only grants anonymization pattern-based queries, so PII cannot be fully disclosed.

Query at Intersection of TLDs

Two types of queries are available: subject-based and pattern-based:

1. At $Super - peer_{\alpha\cap\beta}$, only provides pattern-based queries, at $Super - peer_{\alpha}$ and $Super - peer_{\beta}$ we provide both.
2. A guardian agent in $Super - peer_{\alpha\cap\beta}$ only grants anonymization pattern-based queries, so PII cannot be fully disclosed.

Stratum One Exception:
A Data Owner's Consent

No data Disclosure unless a data owner's consent

$\texttt{Ab1} \rightarrow \texttt{hasPartOf.Condition(Ab1)}$
$\texttt{hasPartOf.Condition(Ab1)}$,

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

Stratum Two Exception:
Without a Data Owner's Consent

Data Disclosure without a data owner's consent

$\texttt{Ab2} \rightarrow \texttt{hasPartOf.Condition(Ab2)}$
$\texttt{hasPartOf.Condition(Ab2)}$,

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

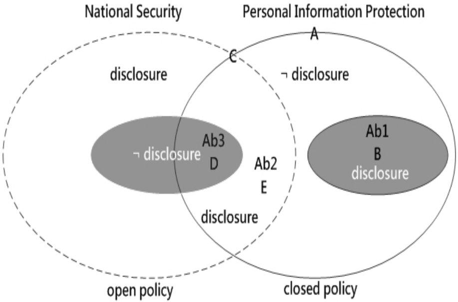
Stratum Three Exception:
Citizen-ships are the Criteria

Deny data disclosing if not a local citizen

$\texttt{Ab3} \rightarrow \texttt{hasPartOf.Condition(Ab3)}.$
$\texttt{hasPartOf.Condition(Ab3)},$

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

A Policy's Exceptions Handling in $SPD_{\alpha\cap\beta}$



Stratified *Datalog*$^\neg$ Rule for Policy Exceptions Handling

COMPLYING WITH TWO TYPE OF LAWS

$\texttt{Request}(?\texttt{r}) \wedge \texttt{hasCondition}(?\texttt{r}, \texttt{Ab1}) \wedge \texttt{Condition}(\texttt{Ab1})$
$\wedge\ \texttt{DomainPolicy}(?\texttt{dmp}) \wedge \texttt{hasCondition}(?\texttt{dmp}, ?\texttt{dmc}) \wedge \texttt{Condition}(?\texttt{dmc})$
$\wedge\ \texttt{isSubsumed}(\texttt{Ab1}, ?\texttt{dmc}) \wedge \texttt{domainPolicyForSPD}(?\texttt{dmp}, ?\texttt{spd})$
$\longrightarrow \texttt{getInTo}(?\texttt{r}, ?\texttt{spd})$

Conclusion

1. A semantic privacy preserving model provides legalized data integration and protection services in semantic cloud.
2. Law-as-a-Service (LaaS) overcomes legal obstacles when Cloud Service Providers (CSPs) intend to deploy their cloud resources and services.
3. Semantic web technologies are applied for semantic legal policy representation to enable data integration and protection.
4. Semantic legal policies, as a combination of ontologies and stratified Datalog rules with negation, are enforced and a semantic legal policy's exceptions are handled through defeasible reasoning.

Conclusion

1. A semantic privacy preserving model provides legalized data integration and protection services in semantic cloud.
2. Law-as-a-Service (LaaS) overcomes legal obstacles when Cloud Service Providers (CSPs) intend to deploy their cloud resources and services.
3. Semantic web technologies are applied for semantic legal policy representation to enable data integration and protection.
4. Semantic legal policies, as a combination of ontologies and stratified Datalog rules with negation, are enforced and a semantic legal policy's exceptions are handled through defeasible reasoning.

Conclusion

1. A semantic privacy preserving model provides legalized data integration and protection services in semantic cloud.
2. Law-as-a-Service (LaaS) overcomes legal obstacles when Cloud Service Providers (CSPs) intend to deploy their cloud resources and services.
3. Semantic web technologies are applied for semantic legal policy representation to enable data integration and protection.
4. Semantic legal policies, as a combination of ontologies and stratified Datalog rules with negation, are enforced and a semantic legal policy's exceptions are handled through defeasible reasoning.

Conclusion

1. A semantic privacy preserving model provides legalized data integration and protection services in semantic cloud.
2. Law-as-a-Service (LaaS) overcomes legal obstacles when Cloud Service Providers (CSPs) intend to deploy their cloud resources and services.
3. Semantic web technologies are applied for semantic legal policy representation to enable data integration and protection.
4. Semantic legal policies, as a combination of ontologies and stratified Datalog rules with negation, are enforced and a semantic legal policy's exceptions are handled through defeasible reasoning.

Future Work

- Exploring defeasible reasoning of a policy's exceptions handling from different hybrid integration of DL-Lite species' ontologies and stratified Datalog rules with negation.
- Exploiting expressive power and computational complexity of semantic legal policy enforcement under different ontologies and rules integration.
- After direct mapping from a RDB's tables to modular ontologies, through fragmentation and encryption techniques to ensure the data protection criteria of outsourcing in the cloud.
- Using tremendous amount of RDB data sets as ontology's data sources to verify sustainability of LaaS.

Future Work

- Exploring defeasible reasoning of a policy's exceptions handling from different hybrid integration of DL-Lite species' ontologies and stratified Datalog rules with negation.
- Exploiting expressive power and computational complexity of semantic legal policy enforcement under different ontologies and rules integration.
- After direct mapping from a RDB's tables to modular ontologies, through fragmentation and encryption techniques to ensure the data protection criteria of outsourcing in the cloud.
- Using tremendous amount of RDB data sets as ontology's data sources to verify sustainability of LaaS.

Future Work

- Exploring defeasible reasoning of a policy's exceptions handling from different hybrid integration of DL-Lite species' ontologies and stratified Datalog rules with negation.
- Exploiting expressive power and computational complexity of semantic legal policy enforcement under different ontologies and rules integration.
- After direct mapping from a RDB's tables to modular ontologies, through fragmentation and encryption techniques to ensure the data protection criteria of outsourcing in the cloud.
- Using tremendous amount of RDB data sets as ontology's data sources to verify sustainability of LaaS.

Future Work

- Exploring defeasible reasoning of a policy's exceptions handling from different hybrid integration of DL-Lite species' ontologies and stratified Datalog rules with negation.
- Exploiting expressive power and computational complexity of semantic legal policy enforcement under different ontologies and rules integration.
- After direct mapping from a RDB's tables to modular ontologies, through fragmentation and encryption techniques to ensure the data protection criteria of outsourcing in the cloud.
- Using tremendous amount of RDB data sets as ontology's data sources to verify sustainability of LaaS.

LaaS System Demo and Q&A

LaaS System Demo. and Q&A

- LaaS System Demo.
- Q&A

LaaS System Demo and Q&A

LaaS System Demo. and Q&A

- LaaS System Demo.
- Q&A

LaaS System Demo(1)

Ent Lab. at NCCU in Taiwan

Home | Peer | SPD | Laas Test | Trust Virutal Domain

- National Taiwan University Hospital
- Center of Disease Control in Taiwan
- National Immigration Agency
- The Taipei Government
- National Security Bureau
- Taipei District Prosecutors Office
- Acer

Center of Disease Control in Taiwan

Welcome! Lin(logout)

Notification Unit	Report Number	disease type	Law Verify	Confirm
National Taiwan University Hospital	2	H1N1	Law Verify	Confirm

α∩β domain

National Taiwan University Hospital

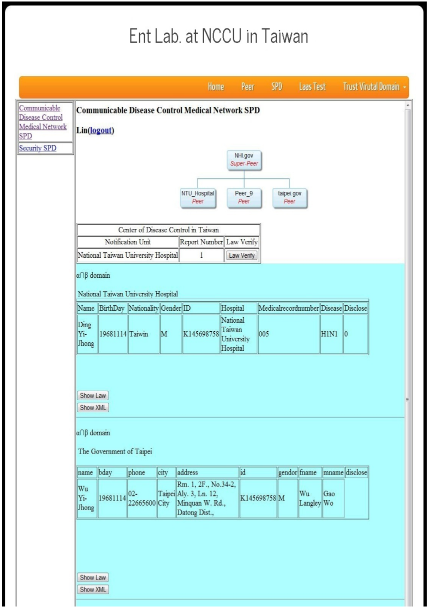
Name	BirthDay	Nationality	Gender	ID	Hospital	Medicalrecordnumber	Disease	Disclose
Ding Yi-Jhong	19681114	Taiwin	M	K145698758	National Taiwan University Hospital	005	H1N1	0

Hide Low

Show XML

	PreventHarm_I	Article	Content
Exception	Personal_Infomation_Protection_Act_C3-3	16	Where it is to prevent harm on the life, body, freedom or property of the Party;
Law	Enforcement_Rules_of_the_Communicable_Disease_Control_Act_16	16	In accordance with regulations of Paragraph 4, Article 39 of the Act, require medical institutions, physicians, or forensic medicine physicians to provide within a definite time, relevant information of patients of communicable diseases;

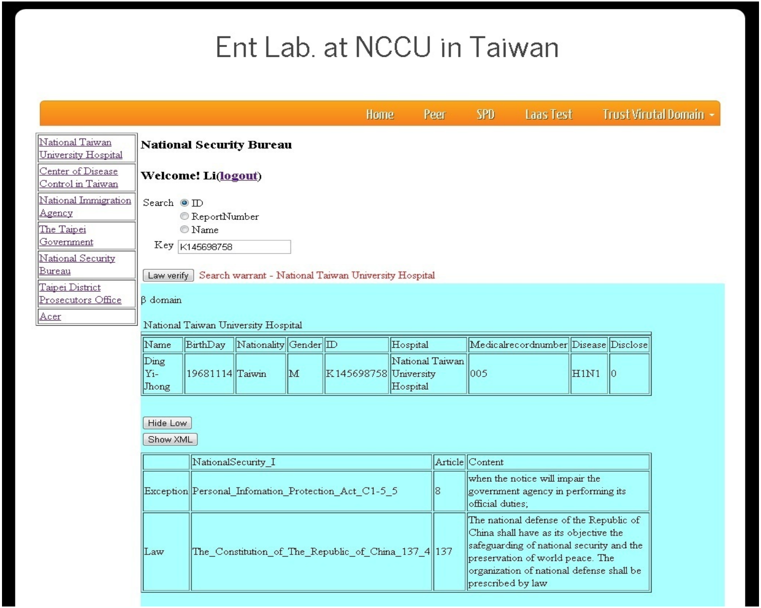
LaaS System Demo(2)



	PreventHarm_I	Article	Content
Exception	Personal_Infomation_Protection_Act_C3-3	16	Where it is to prevent harm on the life, body, freedom or property of the Party;
Law	Enforcement_Rules_of_the_Communicable_Disease_Control_Act_16	16	In accordance with regulations of Paragraph 4, Article 39 of the Act, require medical institutions, physicians, or forensic medicine physicians to provide within a definite time, relevant information of patients of communicable diseases;

	PreventHarm_I	Article	Content
Exception	Personal_Infomation_Protection_Act_C1-5	8	when the notice will impair the government agency in performing its official duties;
Law	Enforcement_Rules_of_the_Communicable_Disease_Control_Act_16_2	16	In accordance with regulations of Paragraph 4, Article 39 of the Act, require medical institutions, physicians, or forensic medicine physicians to provide within a definite time, relevant information of patients of communicable diseases;

LaaS System Demo(3)



Ent Lab. at NCCU in Taiwan

Home | Peer | SPD | Laas Test | Trust Virutal Domain

- National Taiwan University Hospital
- Center of Disease Control in Taiwan
- National Immigration Agency
- The Taipei Government
- National Security Bureau
- Taipei District Prosecutors Office
- Acer

National Security Bureau

Welcome! Li(logout)

Search ◉ ID ○ ReportNumber ○ Name

Key K145698758

Law verify — Search warrant - National Taiwan University Hospital

β domain

National Taiwan University Hospital

Name	BirthDay	Nationality	Gender	ID	Hospital	Medicalrecordnumber	Disease	Disclose
Ding Yi-Jhong	19681114	Taiwin	M	K145698758	National Taiwan University Hospital	005	H1N1	0

Hide Low

Show XML

	NationalSecurity_I	Article	Content
Exception	Personal_Infomation_Protection_Act_C1-5_5	8	when the notice will impair the government agency in performing its official duties;
Law	The_Constitution_of_The_Republic_of_China_137_4	137	The national defense of the Republic of China shall have as its objective the safeguarding of national security and the preservation of world peace. The organization of national defense shall be prescribed by law

[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]Y. J. Hu, W. N. Wu, and J. J. Yang.
Semantics-enabled Policies for Super-Peer Data Integration and Protection.
In *International Journal of Computer Science and Applications (IJCSA)*, 9(1):23-49, 2011.

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

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

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

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

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

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

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

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

[36]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.

[37]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.