cryptography

MDPI

*Article*

# Automating Privacy Compliance Using Policy Integrated Blockchain

**Karuna Pande Joshi [1,*] and Agniva Banerjee [2]**

[1] Department of Information Systems, University of Maryland Baltimore County, Baltimore, MD 21250, USA
[2] Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD 21250, USA; agniv1@umbc.edu
* Correspondence: karuna.joshi@umbc.edu; Tel.: +1-410-455-8775

Received: 1 December 2018; Accepted: 31 January 2019; Published: 5 February 2019

**Abstract:** An essential requirement of any information management system is to protect data and resources against breach or improper modifications, while at the same time ensuring data access to legitimate users. Systems handling personal data are mandated to track its flow to comply with data protection regulations. We have built a novel framework that integrates semantically rich data privacy knowledge graph with Hyperledger Fabric blockchain technology, to develop an automated access-control and audit mechanism that enforces users' data privacy policies while sharing their data with third parties. Our blockchain based data-sharing solution addresses two of the most critical challenges: transaction verification and permissioned data obfuscation. Our solution ensures accountability for data sharing in the cloud by incorporating a secure and efficient system for End-to-End provenance. In this paper, we describe this framework along with the comprehensive semantically rich knowledge graph that we have developed to capture rules embedded in data privacy policy documents. Our framework can be used by organizations to automate compliance of their Cloud datasets.

**Keywords:** Blockchain; Big Data; Semantic Web; privacy policy; ontology; knowledge graph; data compliance

## 1. Introduction

With the increasing adoption of cloud-based services, consumers and organizations are recognizing the need to be able to monitor, in real time, their Personally Identifiable Information (PII) residing on a Cloud service provider's infrastructure. This is especially critical when the sensitive data is shared by providers with their subsidiaries or third-party vendors. Moreover, data could also be accessed by unauthorized transmission of information, leading to an inadvertent breach or leakage. To address concerns about PII data security and privacy, government agencies and regulatory bodies around the world have developed policies and guidelines to secure cloud data.

Cloud providers provide consumers with privacy policy documents, as part of their service contract, that describe their privacy controls in detail along with the regulations that they adhere to. The valid operations (like sharing, deletion etc.) that can be performed on PII datasets are usually described as rules in the privacy policy agreements to which both the cloud service provider and consumers are a signatory. However, these policies often contain legalese jargon that can be hard to comprehend for consumers. Moreover, these text-based documents are not machine processable. Currently, validating the rules embedded in the policy documents with the rule of the data operation requires significant manual effort, and so it is difficult to determine in real-time if a data policy violation has occurred or not. Real time tracking of PII data flow on the Cloud will ensure that any operation on the consumer's data, right from acquisition of the data, it's manipulation or sharing, to

its end-state archival in an organization, along with the validity of the action, can be verified and documented for future audits. This will also facilitate the validation of data operations with data protection regulations.

We believe that Blockchains that can use the power of policy reasoners will address this critical need to be able to track, at real-time, the flow of the consumer's PII data on the Cloud. Our key contribution in this work has been to integrate a machine processable policy framework with permissioned Blockchain to create a novel methodology that will facilitate automatic tracking and auditing of data that is shared among multiple stakeholders including consumers, providers, regulators, and third-party vendors. This methodology uses technologies from the Semantic Web, permissioned Hyperledger Blockchain and Natural Language Processing (NLP)/ Text Extraction. We have also built a system that can parse privacy policy documents and capture the rules, privileges, and obligations governing PII sharing, and also identify and track all data operations which take place. This system will also enable regulatory authorities, cloud service providers and end-users alike to access a transparent, verifiable and immutable ledger containing all data operations along with their validity. In our previous work, we created an integrated methodology to significantly automate the cloud service lifecycle using Semantic Web technologies and Text mining techniques [1–3]. We also developed semantically rich knowledge graphs to capture key elements of cloud SLAs [1] and privacy policies [4]. We have extended this prior work to integrate with permissioned Hyperledger Blockchain to build the LinkShare prototype, briefly described in reference [5], which is based on our novel methodology.

In this paper, we present our methodology in detail and describe the system that we have built to validate our methodology. Section 2 covers the related work in this area. Section 3 describes the underlying technologies that we have used to build our system. Technical details of our methodology are covered in Section 4 and section 5 covers our performance evaluation results. We conclude in the last section.

## 2. Related Work

There has been limited work on integrating Knowledge graphs (or Ontologies) and policy reasoners with Blockchain technologies to authorize and track sharing of data. Kim and Laskowski in reference [6] describe their Ethereum blockchain based proof of concept that integrates with the TOVE [7] traceability ontology. Their implementation is limited because of Ethereum's technical constraints and doesn't include reasoning capabilities. Other researchers who have proposed using Blockchain technologies to manage data privacy have relied on either programmatic or cryptographic solutions that don't incorporate dynamic data sharing policies. Zyskind et al. [8] have proposed a protocol that combines blockchain and off-blockchain storage to construct a personal data management platform focused on privacy. Their approach is restricted to allowing user access based on their digital signatures and does not consider organizational or user policies that can be dynamically changed. Kosba et al. [9] have developed the Hawk system that provides transactional privacy using cryptographic protocols. Their system is designed specifically for smart contracts and cryptographically hides the Hawk program to ensure privacy. Sutton and Samavi [10] have proposed a Linked Data based method of utilizing blockchain to create tamper-proof audit logs.

Researchers have proposed approaches to address data privacy and access control [11–13]. Chen et al. [14] have described a different model for cloud data access called CPRBAC (Cloud-based Privacy-aware Role Based Access Control) model. In the OAuth protocol [15] that is widely used in the industry, companies themselves act as the trusted centralized authority. Such approaches to deal with privacy compliance have typically been role based or attribute based. There also exist ontologies that have been suggested as ways to represent access control concepts, along with legal requirements [16]. Primarily, privacy-preserving methods include differential privacy, a technique that perturbs data or adds noise to the computational process before sharing the data, and encryption schemes that allow running computations and queries over encrypted data. Specifically, fully homomorphic encryption (FHE) schemes allow any computation to run over encrypted data but are currently too inefficient to be widely used in practice.

We believe integrating machine-processable policy representations with Blockchain technologies will significantly aid in automating management of data privacy. As part of this research, we have built the LinkShare system as a proof of concept to illustrate the feasibility of integrating policy reasoners with Blockchain technology. This system was briefly described in reference [4], and we present the details of our system in this paper.

We have also developed a detailed ontology describing the components of the data privacy policies. A preliminary version of this ontology was published in reference [5]. In this paper, we describe the extended and final version of the data privacy policy. While identifying the critical privacy controls that should be specified by the privacy policy documents, we reviewed various standards and guidelines proposed for data security and privacy policy by organizations like the US National Institute of Standards and Technology (NIST)(SP 800-144 and SP 800-53) [17–19] European Union data protection standard [20,21], privacyalliance.org [22], the US Federal Trade Commission [23], and the United States Small Business Administration [24]. Section 4.1 details the Privacy policy knowledge graph/ontology that we have developed.

## 3. Underlying Technologies

### *3.1. Hyperledger Fabric Blockchain*

Blockchain is a peer to peer distributed ledger technology that is being increasingly adopted as an implicitly trustable, confidential, secure and auditable platform [25]. Cryptocurrencies, like BitCoin [26], use blockchain as a publicly verifiable open ledger, and are the most prominent application of this technology. Blockchain also provides opportunities for Cloud-based service providers to automate their business processes. Cloud providers often operate globally and must maintain and engage in transactions with end-users that fully comply with the privacy policies applicable to an individual based on his/her location. By providing regulatory authorities, cloud-based service providers, trusted third parties and end-users with a mechanism for the controlled exchange of sensitive, permissioned data, blockchain technology could improve data sharing and transparency between concerned parties. However, currently, the blockchain protocol does not allow a semantically rich policy reasoner to be implemented.

We have used the Hyperledger Fabric [27], an open source collaborative software, for building our system. It is an advanced model of blockchain fabric, and it is used as a protocol for business to business and business to customer transactions. Record repositories, smart contracts (a decentralized consensus-based network, digital assets, and cryptographic security) are key parts of the Hyperledger. The technology provides a decentralized, transparent, and authenticated platform that applies a consensus-driven approach to facilitate the interactions of multiple entities using a shared ledger. It supports specifying different requirements when competing stakeholders work on the same network. This makes it suitable for implementing our framework since privacy policies contain rules that are usually for Business to business (B2B) or Business to Customer (B2C). The consensus approach used in Fabric is permissioned and voting based. One required parameter is that the nodes in the network must be known and connected and this is not a challenge since all the stakeholders can be easily determined. Our key technical reasons for selecting Hyperledger Fabric include:

1. Fabric is permissioned: The nodes participating in the ledger are linked based on identities provided by the modular membership service provider, which in our case will be the Cloud provider.
2. Fabric nodes have roles: the implementation of Hyperledger Fabric stipulates that the ledger consist of nodes with task-based roles, for example, clients who can submit transaction proposals, peers can validate or execute such transaction proposals, and Ordering Service Nodes, which maintain the total order of the Fabric. These characteristics were needed to make sure the same node which proposes a transaction does not get to validate and execute the transaction. This is imperative to disallow client nodes from executing and validating transactions on their own.
3. Fabric Performance: In Hyperledger Fabric, the delay between transaction proposal, validation and ledger update is similar to the running time of primary-backup replication of data entity in

replicated databases with synchronization through middleware [28]. Low Latency is another reason for us choosing the Hyperledger Fabric. Based on the research by Androulaki et al. [29], for block-sizes under 2 MB, 3 K transactions per second throughput with close to 900 ms latencies had been achieved. For our case, the block-sizes were in the range of 0.5–1 MB (including all PII data fields), and the comparable throughput and latency were 3.5 K transactions per second and 100 ms respectively. As per their research, latency tends to run higher with an increase in block-size beyond 2 MB, but the throughput does not get affected as much.

### *3.2. Semantic Web*

In a virtualized service-oriented scenario, consumers and providers need to be able to exchange information, queries, and requests with some assurance that they share a common meaning. This is critical not only for the data but also for the privacy policies followed by service consumers or providers. While the handling of heterogeneous policies is usually not present in a closed and centralized environment, it is an issue in the open cloud. The interoperability requirement is not just for the data itself, but even for describing services, their service level agreements and their policies for sharing data.

One possible approach to this issue is to employ Semantic Web techniques for modeling and reasoning about services related information. In our prior work, we have successfully demonstrated using Semantic Web technologies to automate cloud services [1] and Service Level Agreements [3]. We have also used them for developing the reasoning and provenance capabilities of our LinkShare system (described in Section 4.3). Semantic Web enables data to be annotated with machine-understandable metadata, allowing the automation of their retrieval and their usage in correct contexts. Semantic Web technologies include languages such as Resource Description Framework (RDF) [30] and Web Ontology Language (OWL) [31] for defining ontologies and describing metadata using these ontologies as well as tools for reasoning over these descriptions. These technologies can be used to provide common semantics of privacy information and policies enabling all agents who understand basic Semantic Web technologies to communicate and use each other's data and Services effectively.

We have implemented semantic reasoner in our system mainly for two reasons: 1. Analytics: Semantic web technologies can help drive analytics on the system. It enables the scope to access, filter, search and analyze privacy policy ontology. It can be used to change the underlying privacy policy ontology in any type of changing requirement. Moreover, it can be used to relate concepts and relationships across the classes of the ontology. 2. It helps drive the reasoning system which flags every data transaction as valid/invalid with respect to the underlying privacy policy. To make sure that only a restricted set of permissible actions (as per the privacy policy) are enforced upon the PII data entities, we have designed a special purpose sub-system within the system which works along-side the blockchain ledger and populates it with transactions marked as valid/invalid. These replace the traditional mining operations, which are extremely costly and performance inhibitive. With this semantic web-based implementation, each piece of data as per the privacy policy has their own set of varying levels of access. The underlying privacy policy governs the set of all permissible actions, and the end-user and the service provider can set/reset the access level classes as per the policy rules and guidelines. The semantic web reasoner handles the validity of these set/reset of access level classes.

## 4. Results and Discussion

As discussed in the introduction, there is an urgent need to be able to track every data operation on a Cloud provider's environment to ensure data compliance and protection. The enterprise and regulatory policies that help manage the privacy and security of the datasets are often text-based and not machine processable. Hence, the current process of ensuring data compliance on the cloud is a time-consuming manual effort. Our key contribution in this work has been to integrate policy framework with permissioned Blockchain technology to create a novel methodology that will facilitate automatic tracking and auditing of data that is shared among multiple stakeholders

including consumers, providers, regulators, and third-party vendors. Figure 1 illustrates the proposed methodology. Machine processable data policies in RDF format are shared among all members of the data sharing network. Each data operation, if approved by these shared data policies, is tracked using Hyperledger Fabric blockchain along with the provenance of the policy. The Blockchain also tracks instances of invalid data operations and the corresponding policy that was violated by the operation. This design allows it to maintain an audit log of all transactions which can be used in real-time to alert data administrators of illegal data shares and/or potential data breaches. We have built the LinkShare system, that has been briefly described in [4], as a proof of concept to illustrate the feasibility of incorporating policy reasoners in Blockchain technology.

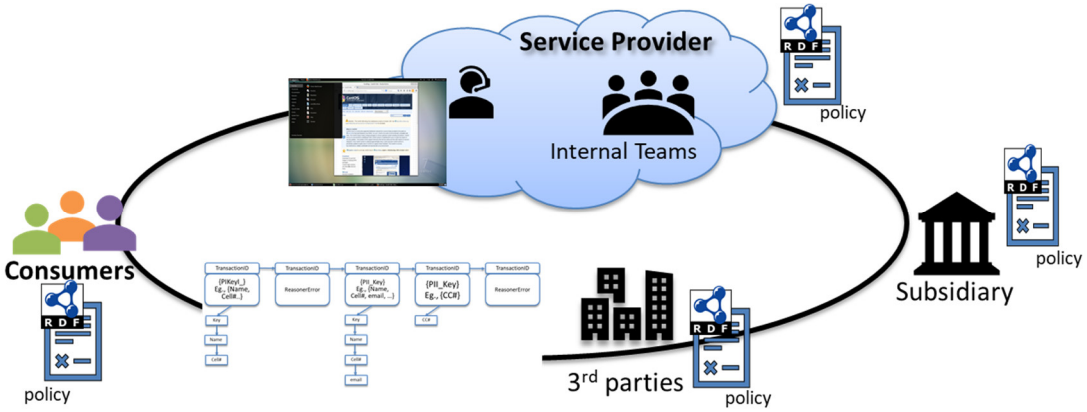


**Figure 1.** Our methodology integrates dynamic policy framework with Hyperledger Fabric Blockchain.

Our design consists of three main phases listed below. Sub-sections 4.1, 4.2 and 4.3 describe the three phases of our design in detail.

1. **Create and populate Knowledge graph**: We used Semantic Web technologies to build a knowledge graph or ontology to capture all the key elements of policy documents. Semantic Web allows us to translate text-based policy documents into machine processable graph datasets. For the first phase of the project, we have concentrated on data privacy policies. Section 4.1 describes this ontology in detail. We are also building knowledge graphs for regulatory policies like EU GDPR and PCI DSS that are available in reference [32]. Using text extraction techniques, we populated our knowledge graph with privacy policies from the ACL COLING dataset [33].
2. **Identify key Data Operations**: We identified the key data operations that would need to be tracked to ensure policy compliance. Section 4.2 describes the various data operations that are monitored and included in our framework design.
3. **Data Compliance BlockChain**: We next built a system using Semantic Web, NLP/Text Extraction and Hyperledger Fabric blockchain to capture each data operation, identified in phase 2, after validating the operation against the policies captured in phase 1. Section 4.3 describes this in detail. Our current implementation is designed for one instance involving all the stakeholders. We are currently extending this work to include multiple consumer and provider organizations sharing the same service.

The work pipeline in the initial stages of our system consisted of ingesting privacy policy documents in the form of text/XML. We have used ACL COLING dataset [33] of privacy policies which consists of around 1011 privacy policies for parsing and knowledge extraction in XML format. For this research, we have considered the privacy policy texts from the year 2014 which was the latest batch of privacy policies available in the dataset. Personally Identifiable Information (PII) such as phone number, address, financial information such as bank account details and credit card details are regularly collected by service providers. The scope of such collection, storage, and distribution of PII are described in the Privacy Policy, which is presented in a textual format to the end-user and to which he/she must be a signatory. The contents of the privacy policy can be challenging for an end-user to understand, and from the service provider's end, an entire system must be designed which

essentially captures the permission, obligations, rules, and regulations stipulated in the privacy policy. Such a system should also make sure that these rules and regulations are adhered to.

### *4.1. Knowledge Graph for Data Privacy Policy*

In this section, we describe in detail the knowledge graph or ontology that we have developed using OWL language to define the range of information that should be included in the Privacy Policy documents. A preliminary version of this ontology was published in reference [34]. On reviewing the privacy policies of leading cloud service providers; we observed that they primarily describe the user data they capture and use and/or share. We compared the various data privacy standards that will be best suited for Big Data applications hosted on the cloud and determined that NIST Special Publication 800-144 [17], that provides guidelines on security and privacy in public cloud computing, and NIST SP 800-53 [18] that listed the privacy controls that are part of the federal cloud computing standards, are best suited for our ontology. These privacy controls are based on the Fair Information Practice Principles (FIPPs) 121 embodied in the Privacy Act of 1974, Section 208 of the E-Government Act of 2002, and Office of Management and Budget policies. This ontology is available in the public domain and can be accessed at reference [35].

#### 4.1.1. Privacy Controls included in Ontology

To build our knowledge graph we concentrated on the following families of privacy control, identified in NIST SP 800-53 [18] and listed below, that are relevant to all organizations and were observed to be part of most of the publicly available privacy policies. Many state laws require web service providers to display their privacy policies and procedures [36]. We did not include families/controls that included policies that were difficult to validate in real time (like the family of accountability, audit and risk management)

A. Authority and Purpose

This family ensures that organizations: (i) identify the legal bases that authorize a particular personally identifiable information (PII) collection or activity that impacts privacy; and (ii) specify in their notices the purpose(s) for which PII is collected.

- Authority to Collect: The service provider should determine and document the legal authority that permits the collection, use, maintenance, and sharing of personally identifiable information (PII), if required by regulatory and compliance bodies.
- Purpose Specification Control: The organization describes the purpose(s) for which personally identifiable information (PII) is collected, used, maintained, and shared in its privacy notices.

B. Transparency

This family ensures that organizations provide public notice of their information practices and the privacy impact of their programs and activities.

- Privacy Notice: The organization

a. provides notice to the public and to individuals regarding (i) its activities that impact privacy, including its collection, use, sharing, safeguarding, maintenance, and disposal of PII; (ii) authority for collecting PII; (iii) the choices, if any, individuals may have regarding how the organization uses PII and the consequences of exercising or not exercising those choices; and (iv) the ability to access and have PII amended or corrected, if necessary;

b. Describes: (i) the PII the organization collects and the purpose(s) for which it collects that information; (ii) how the organization uses PII internally; (iii) whether the organization shares PII with external entities, the categories of those entities, and the purposes for such sharing; (iv) whether individuals have the ability to consent to specific uses or sharing of PII and how to exercise any such consent; (v) how individuals may obtain access to PII; and (vi) how the PII will be protected; and

c. Revises its public notices to reflect changes in practice or policy that affect PII or changes in its activities that impact privacy, before or as soon as practicable after the change.

- Dissemination of Privacy Program Information:

The organization ensures that the public has access to information about its privacy activities and is able to communicate with its Chief Privacy Officer (CPO), and ensures that its privacy practices are publicly available through organizational websites or otherwise.

C. Data Minimization and Retention

This family helps organizations implement the data minimization and retention requirements to collect, use, and retain only PII that is relevant and necessary for the purpose for which it was originally collected.

- Minimization of personally identifiable information control: The organization -
  a. Identifies the minimum personally identifiable information (PII) elements that are relevant and necessary to accomplish the legally authorized purpose of collection
  b. Limits the collection and retention of PII to the minimum elements identified for the purposes described in the notice and for which the individual has provided consent

  This control also recommends using anonymization and de-identification techniques when using such data to minimize the risk of disclosure.

- Data Retention and Disposal Control: Organization -
  a. Retains each collection of personally identifiable information (PII) for an organization-defined time period to fulfill the purpose(s) identified in the notice or as required by law;
  b. Disposes, destroys, erases, and/or anonymizes the PII, regardless of the method of storage, in a manner that prevents loss, theft, misuse, or unauthorized access; and
  c. Uses organization-defined techniques or methods to ensure secure deletion or destruction of PII (including originals, copies, and archived records).

- Minimization of PII used in testing, training, and research control: The organization:
  a. Develops policies and procedures that minimize the use of personally identifiable information (PII) for testing, training, and research; and
  b. Implements controls to protect PII used for testing, training, and research.

D. Individual Participation and Redress

This family addresses the need to make individuals active participants in the decision-making process regarding the collection and use of their personally identifiable information (PII). By providing individuals with access to PII and the ability to have their PII corrected or amended, as appropriate, the controls in this family enhance public confidence in organizational decisions made based on the PII. The controls in this family include -

- Consent Control: The organization:
  a. Provides means, where feasible and appropriate, for individuals to authorize the collection, use, maintaining, and sharing of personally identifiable information (PII) prior to its collection;
  b. Provides appropriate means for individuals to understand the consequences of decisions to approve or decline the authorization of the collection, use, dissemination, and retention of PII;
  c. Obtains consent, where feasible and appropriate, from individuals prior to any new uses or disclosure of previously collected PII; and
  d. Ensures that individuals are aware of and, where feasible, consent to all uses of PII not initially described in the public notice that was in effect at the time the organization collected the PII.

- Individual Access Control: The organization:
  a. Provides individuals the ability to have access to their personally identifiable information (PII) maintained in its system(s) of records; and

b. Publishes rules and regulations governing how individuals may request access to records maintained in a Privacy Act system of records.

- Redress Control: The organization:
  a. Provides a process for individuals to have inaccurate personally identifiable information (PII) maintained by the organization corrected or amended, as appropriate; and
  b. Establishes a process for disseminating corrections or amendments of the PII to other authorized users of the PII, such as external information-sharing partners and, where feasible and appropriate, notifies affected individuals that their information has been corrected or amended.

E. Use Limitation

This family ensures that organizations only use PII either as specified in their public notices, in a manner compatible with those specified purposes, or as otherwise permitted by law. We include the following controls and features in our knowledge graph

- Internal Use: The organization uses PII internally only for the authorized purpose(s) identified in the Privacy Act and/or in public notices.
- Information Sharing with Third Parties: The organization:
  a. Shares PII externally only for the authorized purposes identified in the Privacy Act and/or described in its notice(s) or for a purpose that is compatible with those purposes;
  b. Where appropriate, enters into Memoranda of Understanding, Memoranda of Agreement, Letters of Intent, Computer Matching Agreements, or similar agreements, with third parties that specifically describe the PII covered and specifically enumerate the purposes for which the PII may be used.

#### 4.1.2. Privacy Ontology Classes

The main classes of the ontology are illustrated in Figure 2. Referring to the NIST guidelines on cloud privacy [17] and PII information [19], we have identified the key components of a privacy notice that are defined as object properties in the main **Privacy Policy class**. The numbers in the brackets indicate the relationship with the class of the functional property. Therefore, each privacy policy should have one instance describing the collection purpose and data protection controls; privacy policy should have at least one instance of consumer consent and Access to own PII, and so on. The main sub-classes are

1) **Collection Purpose**: This class captures the purpose and scope of data collection and the limited use that the data will be subjected to. It also contains information about the actions that will be taken to transform the data, which can include combining it with other datasets or aggregating/summing the data. The policy document should also specify the duration the data will be managed by the data collector and the contact details of the provider's Chief Privacy Officer (CPO).

2) **PII Data Collected**: This class identifies key attributes that comprise personally identifiable information. These include personal details like names, contact information, like address, phone numbers, identity numbers and identity characteristics. These are illustrated in detail in Figure 3. Other PII data includes employment, medical, financial and education details of a person. To identify the key properties of these classes, we referenced the NIST special publication 800–122. [19]

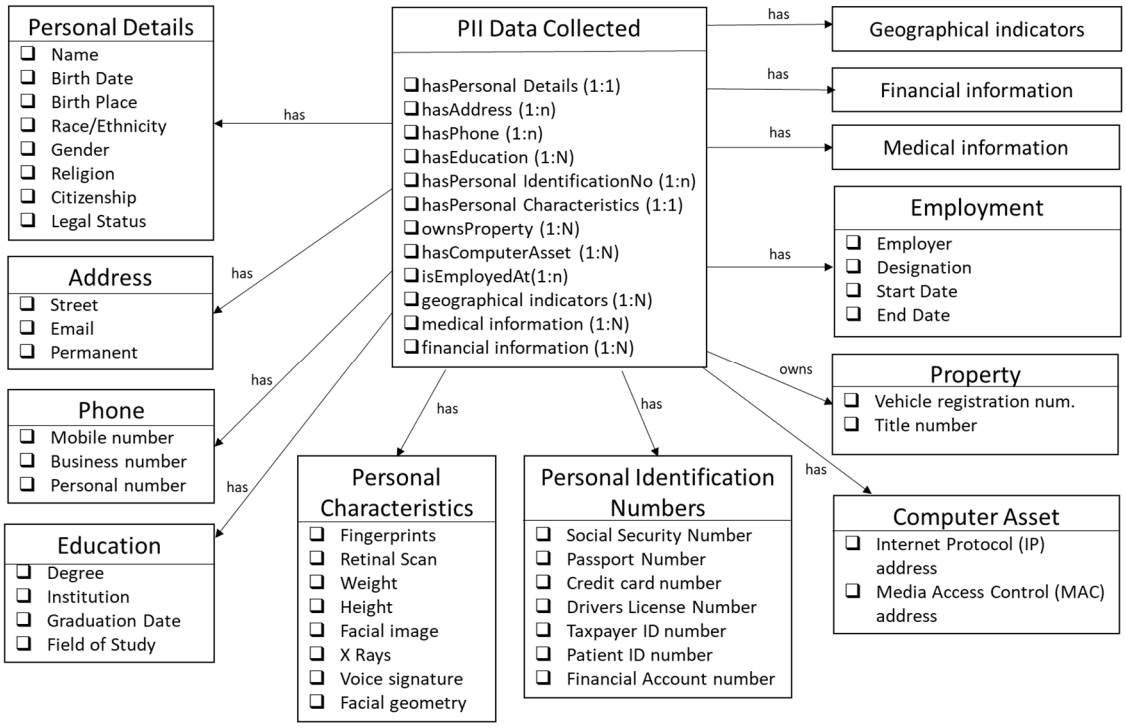


**Figure 2.** High Level Knowledge graph describing components of data privacy policy.

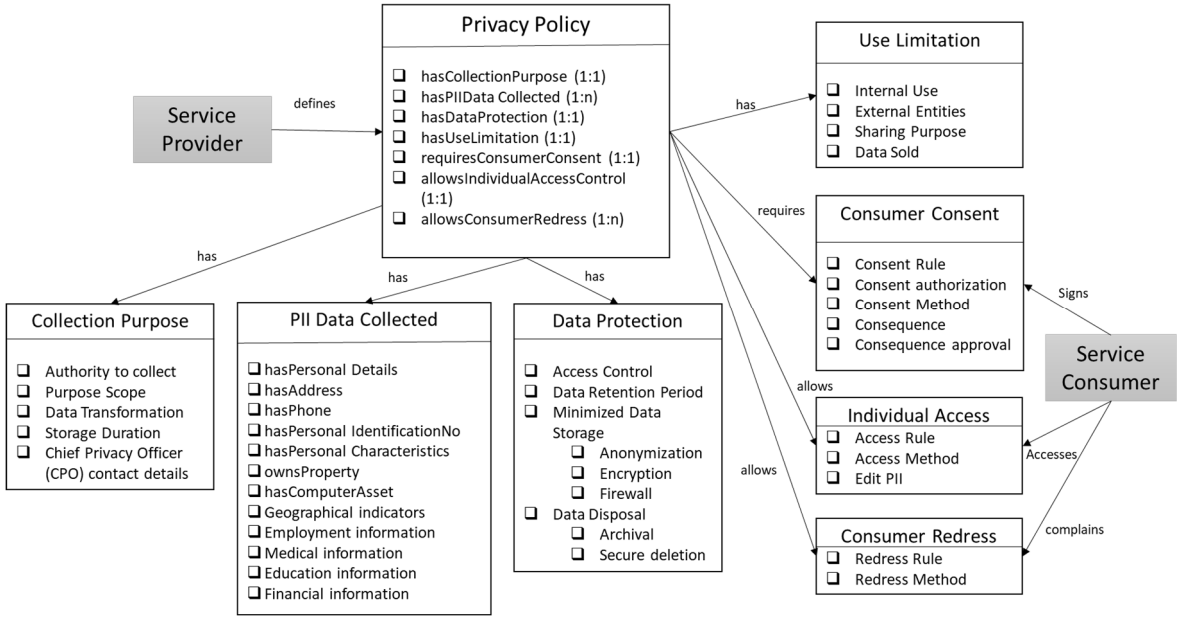


**Figure 3.** Details of the PII Data classes.

Each privacy policy instance may have one or more instances of PII data associated with it, but the number of instances should be small. We allow multiple instances of PII to accommodate data versioning and allow keeping old values of PII even when a new value is added. For instance, a consumer may change their primary address associated with an e-commerce site; the site vendor could retain the previous address of the consumer in a separate instance for internal analysis of consumer behavior. Alternatively, the provider might want to change the PII dataset collected by their service but retain the same collection purpose and data protection policies and so would have multiple instances of just the PII Data Collected class.

As part of our ongoing work, we are linking this ontology with other existing ontologies in the public domain. For instance, the geographical indicators will be linked with the W3C Geospatial Ontologies [36], financial information will reference EDM Council's FIBO [37] financial ontology; the

medical information class will reference existing medical ontologies available at openclinical.org/ontologies.html, etc. For such classes, a single block is displayed in Figure 3.

3) **Data Protection**: This class includes properties pertaining to data access control, retention, disposal and minimized storage controls that should be in place. In our previous work, we have developed OWL ontologies for Role-based access control [38] and attribute-based access control [39] which we plan to integrate with this privacy policy ontology. As part of our planned work, we will also incorporate other publicly available data protection ontologies.

4) **Use Limitation**: This class includes the details of the internal and external entities with whom the PII data will be shared. It includes the purpose of sharing this data and information about whether the data will be sold to external entities.

5) **Consumer Consent**: The consumer's consent should be obtained whenever PII data is captured by the provider. The consent to share the data should be explicitly mentioned. The consent method – signature, agreement, etc. should be specified. This class also contains details of the consequence of the consent and approval of the consequences by the consumer.

6) **Individual Access**: The consumer should be able to access their PII that is maintained by the provider. The access method should be clearly specified in the privacy document.

7) **Consumer Redress**: This includes the method by which the consumer will be able to correct or amend inaccurate PII data maintained by the provider.

### *4.2. Data Operations*

End to end provenance can only be achieved when there are provenance records for both data operations and decisions made by the system. In any decision-making system, it is of prime importance to be able to validate the decision-making process. Such provenance can be used to determine the reliability and quality of the decision-making system. Since the system reasons and then comes to a decision regarding the validity of a certain data transaction, it is thus imperative that it incorporates provenance records. When presented with a questionable decision, such capability will allow regulatory authorities, service providers and end-users alike the capability to seek further validation from the provenance. Also, such provenance capabilities induce trust.

Concern over automated and algorithmic decision-making systems are increasing. One way to address such an issue is by exposing the decision-making pipeline. The flow of input and the decisions the input is processed upon within the system can be exposed to an extent allowed by the system. In this regard, provenance has been achieved by employing Semantic Web reasoning, which validates the request based on the underlying privacy policy ontology. By capturing the chain of data-flow and the associated validation undertaken by the reasoner and by recording the validation steps taken by the reasoner, we have made the system accountable, since these decision-making trails can assist in auditing and investigating the decision-making process.

Since our system is primarily a PII data transformation ledger, it is also imperative for us to capture data provenance. We are maintaining the same by recording everything that happens to a data entity for it to come to its current form. This acts as a form of contextual resource metadata. The data transformation actions captured and recorded by the system can be defined as follows: 1. Data Acquisition: when a new party comes into an agreement, 2. Data Generation: generating or deriving new data points from existing data points such as deriving age from date of birth, 3. Data Manipulation: Any and all sorts of data operation which updates or changes a specific data entity, 4. Data Distribution: sharing the data amongst one's own subsidiaries, and amongst trusted 3rd parties will be considered as an example of data distribution. By capturing such transaction operations, we maintain a traceable ledger of data operations. We describe these data operations that are captured by our system in further detail below:

1. **Data Acquisition**: In a cloud-computing environment, we define data acquisition by the transactions that create new data points that are directly provided by the user and stored by the system. These new data points are created only when a new user comes to an agreement with the cloud service provider and its associated business affiliates and trusted third parties. The transactions are between the cloud-based service provider and the end-user, where the end-user

provides the data and the service provider stores this data. The data points captured by the system must fall under the purview of the privacy policy.

2. **Data Generation**: When a transaction uses existing data points or conjures new data points from them, it is treated as an instance of data generation. The generation of new data points must come from the existing data entities that were provided by the user. For example, generating Age from date of birth (which is provided by the user), or generating user-class (such as credit-risk) from a combination of data entities (such as credit score and annual income). These new data points generated by the system must fall under the purview of the privacy policy that the end-user and the owner of the system that generated the data is a signatory to. The transactions are usually undertaken by the cloud-based service provider, trusted 3rd parties and subsidiaries of the service provider.
3. **Data manipulation**: This refers to any transaction which acts upon any previously stored data and morphs the data entity permanently. Examples of such transaction would be changes in credit score, home address, phone number, etc. These changes can be performed by any of the signatories under the privacy policy.
4. **Data distribution**: Data distribution on the cloud includes all transactions that result in the sharing of the user data by the service provider with other signatories in the privacy policy. This kind of transaction mostly deals with the sharing of data amongst various parties which are bound by the same privacy policies. For example, a cloud-based service provider (such as Instagram) might share a user's browsing pattern with another service provider (such as an advertising service).

Regardless of the validities of these data transactions, these will be recorded by our blockchain based ledger system, in order of execution. This temporal aspect adds value to the records, in a way that can be used for further analysis of the data lifecycle. Using a blockchain based data provenance system is especially useful since it alleviates technical challenges such as having a unified, verifiable provenance traces, and it formally guarantees that the ledger will satisfy immutability of the records.

### *4.3. LinkShare: Data Compliance BlockChain*

In this phase of the design, we developed the LinkShare system as a proof of concept for our approach of integrating policy reasoners with Blockchain technology to automate cloud data compliance. We used Semantic Web Technologies, NLP techniques and Hyperledger Fabric for capturing each data operation, identified in phase 2, after validating the operation against the policies captured in phase 1. Data is immutable once uploaded. Accordingly, there is no need to decompose tables to reduce redundancy and achieve integrity. This system has been briefly described previously in [4].

The system consists of 3 distinct entities: UserBase, PolicyTree, and BlockchainLedger. The UserBase consists of all stakeholders who share the policies and potentially the data, including service providers, and end-users. Based on the policy attributes stored in the ontology, a UserBase can either access, share or contribute to the ledger. The access right of an individual component of the UserBase is completely determined by the underlying policy tree. The PolicyTree, which is the populated knowledge graph (see Section 4.1), is maintained by the Service-Provider. The PolicyTree is built by ingesting a privacy policy in its textual format, and then populating the underlying ontology with the extracted regulations, permissions, obligations, etc. PolicyTree and BlockchainLedger form the backbone of the system. The PolicyTree, or ontology, defines all required access control mechanism and privacy policy.

The Privacy and access controls are included in the ontology itself. The PolicyTree obligates the service provider to determine and document the legal authority that permits the collection, use, maintenance, and sharing of personally identifiable information (PII), as required by regulatory and compliance bodies. Also, it obligates the end-user and the service provider alike to document purpose(s) for which personally identifiable information (PII) is collected, used, maintained, and shared. The higher level contains legislative requirements for data protection and policies related to

operational requirements. The lower levels contain access control relations and policies for handling personal data.

For example, the PolicyTree graph can contain instances of the Data, AccessControl and Purpose classes describing how to consume and store data, how to contain access and sharing the data, and how to classify the reason behind accessing, storing or sharing the data. Drilling down further on Data and AccessControl, we can have granular nodes about individual data points that can be collected, such as Name, Address, Birth date, etc., and the relation between an individual user of the system and such individual data points is controlled by AccessControl relation, which specifies whether a node can be accessed based on relations such as *IsDataOwner*, *IsDataController*, so on and so forth. The individual data points such as Name, DOB, Address, etc. are specific to individual users, and the relations are universal.

Privacy-aware data access policy cannot be easily achieved by traditional access control models. The first reason is that traditional access control models focus on who is performing which action on what data object, while privacy policies focus on the context of data object usage. We have used a context-centric access control model in our design. For instance, consider service providers Netflix, Amazon, Trusted Third Parties, and an end-user User1. All four will be part of the UserBase. The relations for the individual data points such as Name, Address, DOB, *IsDataOwner*, *IsDataController*, *IsDataSharable,* etc. can only be set by the end-user at the time of coming into an agreement with the Service Providers, and these relations can be reset only at the end user's behest. Every time any of the concerned parties (service providers, trusted third parties) engages in a transaction, it will be recorded at the end of all the individual data points for that transaction. For example, if Netflix wants to send Amazon Name, ZIP and CreditCard, a new transaction will be recorded, with (Name, ZIP, CreditCard) and (*IsSharable*, *IsDataRequested*, *IsSensitiveData*) requirements to be fulfilled by individual relations between service providers and the individual data points. Every such data point that has been deemed shareable by the end user becomes added as part of a hashed new node at the end of that specific data-point chain, and only then the transaction is passed.

It is to be noted that this tree does not contain any data pertaining to the user. The PolicyTree acts as an access control system to the transaction ledger. This approach not only adds granularity at Service-provider to End-User data, but it also acts as a verifiable, secure way to comply with any transaction which requires the use of private information. For the end-user, it acts as an assurance that any data that he/she does not want to be share, will not be shared. Meanwhile from the service provider's point of view, it acts as a verifiable, secure ledger to facilitate, verify or enforce the privacy policy requirements.

#### 4.3.1. Key Modules of the LinkShare system

We have utilized Semantic Web, Natural Language Processing (NLP) and Hyperledger [27] based ChainCode to semi-automate the process of linking and sharing end-user data across businesses based on privacy policies. Data is immutable once uploaded. Accordingly, there is no need to decompose tables to reduce redundancy and achieve integrity. We identified key stages in handling the process and broke it down into the following interdependent modules in the system. (Figure 4)

1. **Ingesting Privacy Policy**: The starting point of the system happens through ingestion of the privacy policy in a textual format. This module extracts information from the privacy policy dataset by using NLP techniques. Primarily, two kinds of information are extracted: a. Regions of Interest (ROI) and b. Sub-class relation. It was noticed by manually inspecting the extracted ROIs that there were significant overlaps between ROI extracted for the super-classes. The overlaps were calculated by passing every tuple of ROI TF-IDF (Term frequency- Inverse Documents Frequency) vector for a superclass (e.g. Consumer Consent) sentence with every other TF-IDF vector of sentence extracted for rest of the super-classes (e.g. Use Limitation) to find string overlap over a certain threshold (0.2 in our case). From these term descriptions, specific relations were extracted to populate the subclasses - consumer consent, use limitation, and data

protection in the ontology (see Figure 2). Augmentations include adding Part of Speech (POS) tags and generating parse trees for each individual line in the privacy policy. These augmentations are read along with the actual text, and specific rules are used to extract key terms and definitions, and from the specific regions of interest modalities pertaining to individual subclasses were extracted using deontic/modal logic. This extracted information is used to populate the underlying ontology.

To this end, every sentence is individually parsed and are POS tagged. The extraction pattern used for determining a term of interest and its associated definition is an "IS-A" (inheritance) relationship. The scope of the term definition extraction is further narrowed down by an inheritance relationship. These extracted terms, definitions, permission and obligations form the basis of generating the partially filled base privacy policy ontology.

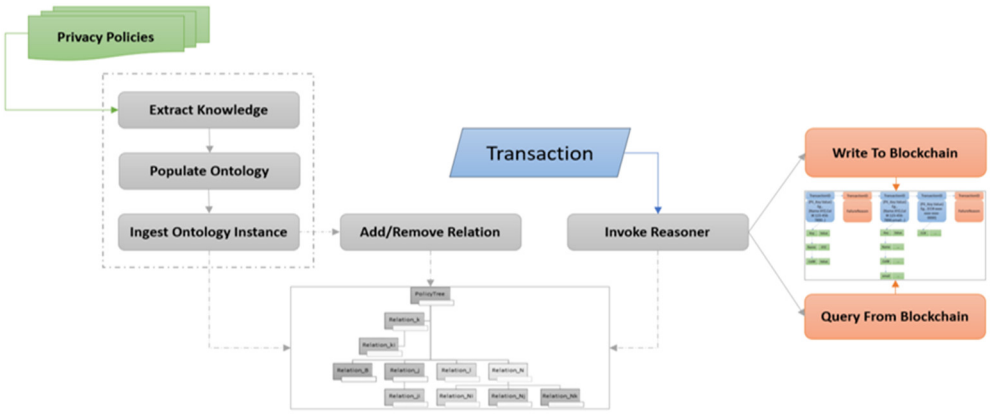


**Figure 1.** LinkShare System Architecture consists of six key processes.

2. **Parsing Ontology**: The privacy policy ontology (described in Section 4.1) is first consumed through a module designed to import OWL 2.0 ontologies in the OWL/XML format. The module can be directly used to load an ontology, either from the internet or from the local repository. Once loaded, it accesses ontology classes, performs automatic classification of classes and instances of the ontology, creates new instances / individuals and stores them for further manipulation in a PolicyTree. We have included the main sub-classes of the privacy policy ontology described in Section 4.1 in our design. Properties that were added to Individual Access and Use Limitation classes to broaden the scope of this class from reference [36] are *IsSharable*, *IsDataRequested*, *IsSensitiveData*, which act as a check for all partaking members of the UserBase, end-users and service providers alike.

3. **Add/Remove relations** based on Privacy Policy specifications: The next module handles the task of adding or removing relations or processing further updates to the PolicyTree created in module 2 described above. Our design assumes that only a service provider can start the process of creating the PolicyTree and add further relations onto it. UserBase can only access and update its permissions. Once the privacy policy has been parsed and uploaded, this module lets the policy be modified based on the user preference, manipulate ontology classes, instances and properties transparently. Also, a new property can be created by sub-classing the Property class, and an existing property can be modified. 'Domain' and 'Range' properties can be specified for the Property as well. A relation is a triple (subject, property, object) where the property is a Property class, and subject and object are instances which are subclasses of the 'Domain' and 'Range' defined for the property class. These relation triples can come from the triples extracted from the privacy policy text. Once the user-preferred relations are created, and instances are made, the ontology is passed onto the next step. It is always possible to come back to this module from the next stage in the system.

4. **Invoke Reasoner**: Upon the execution of any transaction, the Reasoner module is executed, which verifies whether the data units used by the current transaction are not in violation of any rules on the PolicyTree about the concerned user. Based on the underlying privacy policy and the PII fields requested, one of the three conditions are handled by the reasoner:

    1) *Data not sharable*: If the PII data-fields requested in the current transaction do not adhere to the privacy settings by the end-user, the current transaction will be deemed as "Failed" and the *blockchainWrite* method will be invoked.

    2) *Data fully sharable*: If the PII data-fields requested in the current transaction adhere to the privacy settings by the end user, the current transaction would be deemed as "Success," and the respective PII fields will be stored in the blockchain.

    3) *Data partially sharable*: There can be times when the PII fields requested in the transaction partially adhere to the underlying privacy policy. For example, consider a transaction request from Netflix to Amazon, with FirstName, LastName, ZIP, CreditCardNumber and Address PII fields. If one or more PII fields were deemed unsharable by the end-user, then by default the system will regard the transaction as "Failed".

    If the reasoner succeeds, *blockchainBranchWrite* methods are called with *TransactionID* and {Personally Identifiable Information Field - Value}. Otherwise, the *blockchainWrite* is called with TransactionID and ReasonerError.

5. **Write to Blockchain**: This is the module that initializes and invokes the functions in Fabric ChainCode. As illustrated in Figure 5, whenever the blockchain is modified, one of the two methods are called based on the result of the Reasoner: *blockchainBranchWrite* or *blockchainWrite*. *BlockchainBranchWrite* stores the successful transaction as {*TransactionID* - {key-value}} → {key - value}, that is, it creates a main block with *TransactionID* as the key, and hashed PII fields with their corresponding values the value part of the block. The block will always have a branching block which contains PII fields and their corresponding values. The hash of this block can act as the verifiable value for the main block since any modification in the key-value pair would change the hash of the main block. On the other hand, with the failure of a transaction, *blockChainWrite* method is called, which stores the failed transaction with *TransactionID* and *ReasonerError* as a separate block in the blockchain (see Figure 5).

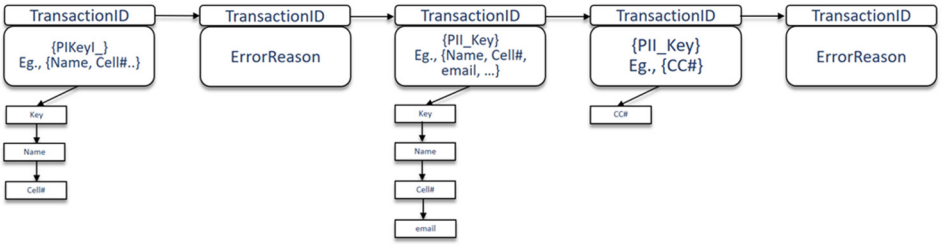


**Figure 5.** Blockchain Structure Diagram.

We have primarily integrated the Linkshare system with Fabric ChainCode by using the "package" and "initialize" commands to initialize and set up the ChainCode and the "invoke" transaction to call a function to query the ontology for confirmation (like whether it allows *isSharable*, *isSensitiveData*, etc.) of every transaction before adding to the blockchain. Section 5.2. illustrates this with an example.

6. **Query Blockchain**: To query the blockchain, we created a separate module which accepts a *TransactionID* and fetches the block pertaining to the transaction after verifying the Ontology using the *GetState* function. Since the blockchain is stored in such a way that any piece of information is directly or indirectly linked with the Transaction it was part of, it is of vital

importance that the Query uses *TransactionID* as part of the Query Key term. This module takes input *TransactionID* for the current user or Service-Provider that is initiating the query, and the block data is available to User/Service-Provider only if they have permission to view/share the results of the Transaction.

## 5. System Evaluation

### *5.1. Extraction Results*

The system pipeline starts with the ingestion of a textual privacy policy. To that end, results of the extraction of key terms and regions of interest about consumer consent, external sharing and data protection from 1011 privacy policies from ACL COLING dataset [33] are detailed below.

The regions of interest extracted from the privacy policies were parsed using grammatical expressions. Amongst the regions of interest extracted, some of the sentences directly contributing towards the specific classes are listed below. Analysis of the overlap and subsequent hierarchy will follow:

- Consumer Consent: "If you are visiting the Services from outside the United States your data will be transferred to and stored in our servers in the U.S. By using the Services, you consent to our collection and use of your data as described in this Privacy Policy."
- External Sharing: "By using the Services or providing us with any information you consent to the transfer and storage of your information including Personal Information to registered third parties as set forth in this SEA Privacy Policy."
- Data Protection: "The data protection laws in the United States may differ from those of the country in which you are located, and your Personal Information may be subject to access requests from governments courts or law enforcement in the United States according to laws of the United States."

It is interesting to observe that the regions of interest extracted by the sub-module overlapped in scope with other classes in ontology, for example, Data protection region of interest shared its scope with External Sharing, in turn, shares its scope with Consumer Consent. Figure 6 shows the ROI extracted across the classes, and it must be noted that the scopes overlap.

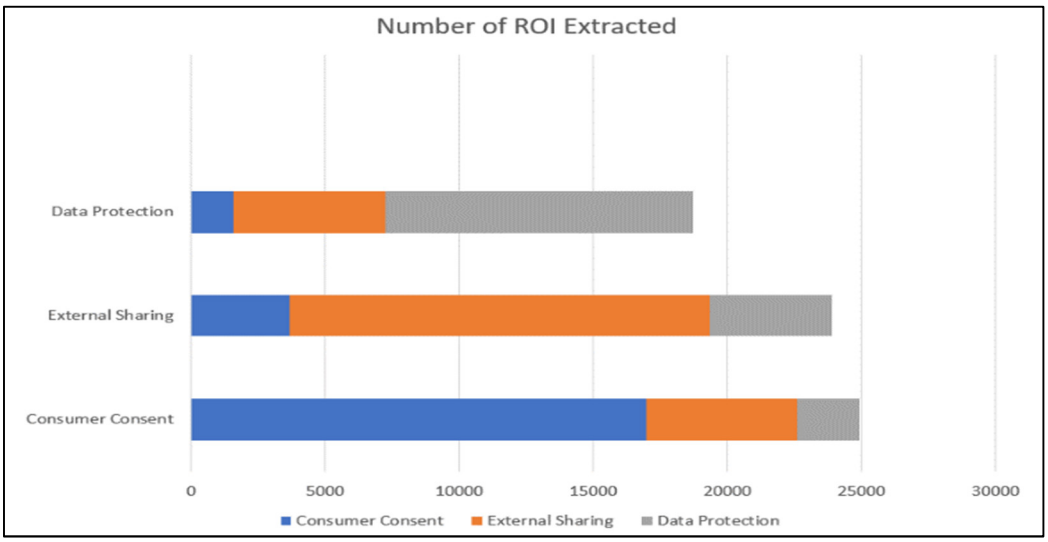


**Figure 6.** Regions of Interest Extracted.

The ROIs extracted were processed further to extract modalities for the subclasses about individual subclasses for each superclass (e.g., Access Control subclass for Data Protection). The modalities extracted for each subclass under the individual superclass is illustrated in Figure 7.

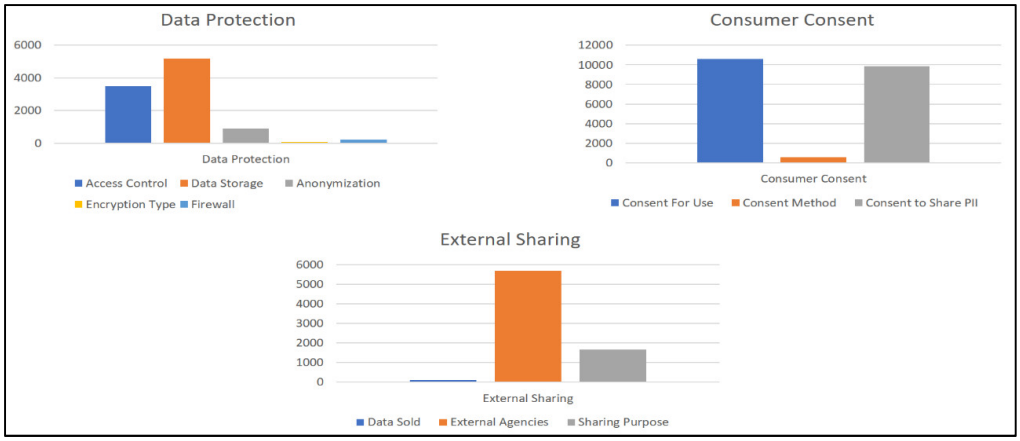


**Figure 7.** Number of Modalities Extracted.

## *5.2. Transaction Processing*

To evaluate our system, we considered the three possible scenarios when a data transaction is validated through the system. Based on the privacy settings of the individual and based on the permissible policies extracted from the privacy policy, the following scenarios can be possible: registered non-institution users in the blockchain. The following parameters were set:

(i) **Data sharing is permitted**: Here, depending upon the PII fields present in the data transaction, and based on the policies that allow such PII to be a valid part of the data transaction, the reasoner will deem the data transaction to be valid and proceed to perform Write operation on blockchain with the PII fields and transaction ID. For example, consider a use case of NetFlix acquiring a new customer {PII fields: FirstName, LastName, Date of Birth and Address}, the execution of this scenario in the prototype is shown in Figure 8.

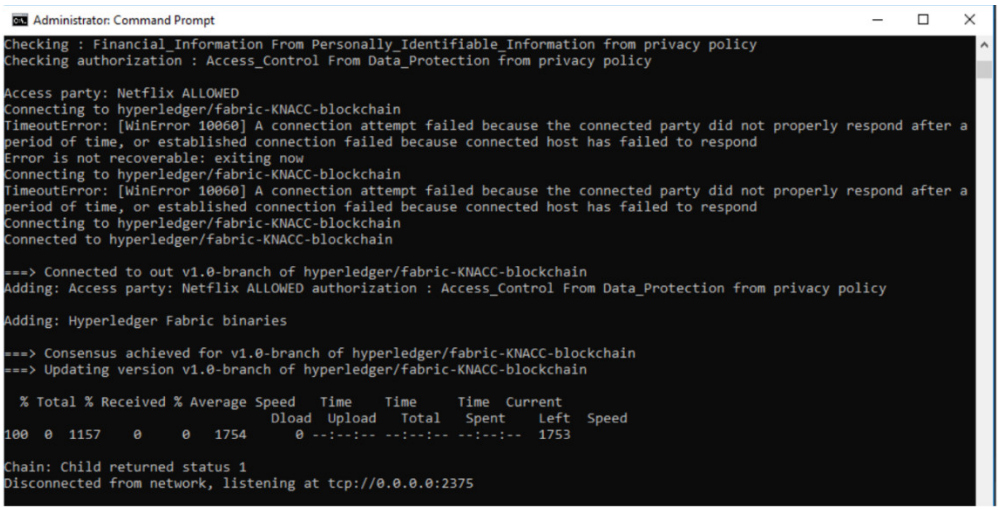


**Figure 8.** System Output when PII Sharing is permitted.

(ii) **Data sharing is partially permitted**: Here, depending upon the PII fields present in the data transaction, and based on the policies that allow such PII to be a part of the data transaction but not wholly involved, the reasoner will still deem the data transaction to be valid since it is the default selection and proceed to perform Write operation on blockchain with the PII fields and transaction ID. E.g., consider a use case of sharing some PII data by the service provider (Netflix) with trusted 3rd party {PII fields: FirstName, LastName, CreditCardNumber}, the execution of this scenario in the prototype system is shown in Figure 9.

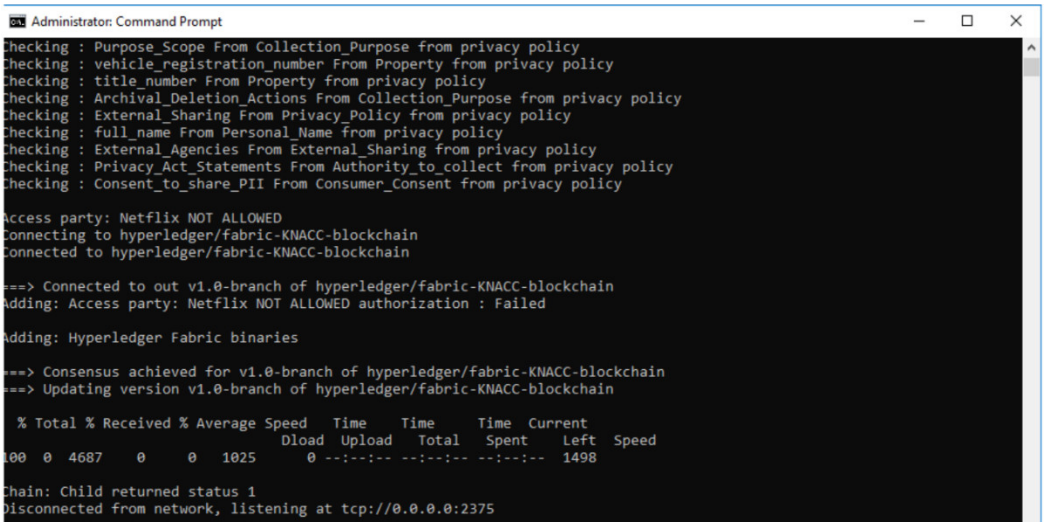


**Figure 9.** System Output when PII sharing is partially permitted.

(iii) **Data sharing is not permitted**: Here, depending upon the PII fields present in the data transaction, and based on the policies that do not allow such PII fields to be a valid part of the data transaction, the reasoner will deem the data transaction to be invalid and proceed to perform a Write operation on blockchain with the transaction ID and FailureReason. This reason acts as part of the Provenance of the system. E.g., for a use case of sharing PII data by the service provider (Netflix) with trusted 3rd party, the execution of this scenario is not permitted by the prototype system as shown in Figure 10.

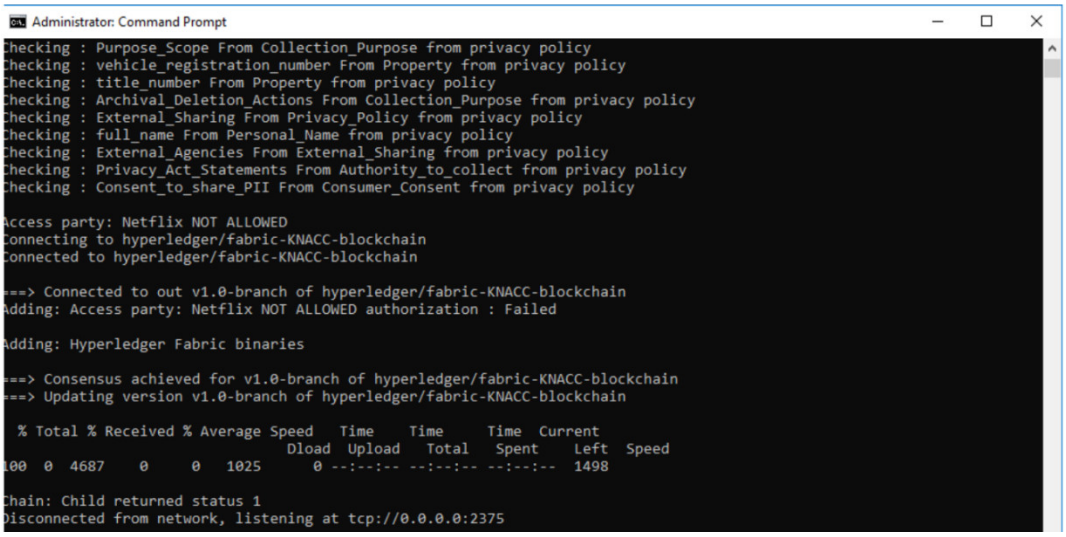


**Figure 10.** System Output when PII Sharing is not permitted.

### *5.3. Performance*

For performance analysis of the proposed blockchain framework, various parameters were measured and evaluated. Small-scale scenarios were considered with 10 nodes. In each scenario nodes were split into two sets: Service Providers, i.e., providers of privacy policy resources, resource requesters which execute smart contracts to perform privacy policy-based transactions; End-Users or registered non-institution users in the blockchain. The following parameters were set:

(i) experiment duration: 100 s: this is set to take into account the request generation, reasoner result and write to blockchain time;

(ii) the Service Providers/End-User ratio: 1:1 – current implementation;

(iii) each Service Providers registered 1 randomly generated End-User;

(iv) each Service Provider sent a randomly generated transaction to write request with Key-Value pair and

(v) each End-User sent a new randomly-generated query request every 10 s.

In case of consuming privacy policy, pass/fail was determined by how robust the system is in handling RDF and OWL-based privacy policies; Add/Remove relations were run through adding viable and conflicting relations, and hence Reasoner shares similar result. Write to blockchain reflects the number of illegal and legal transactions it handled. Query blockchain shows the actual pass/fail numbers, i.e., out of 10 queries, it could successfully handle 4. ChaincodeInvokeOrQuery invokes or queries the chaincode and if successful, the INVOKE form prints the ProposalResponse to STDOUT, and the QUERY form prints the query result on STDOUT. The absence of standard output or error in connecting to running blockchain was considered as Fail. We can see that it is during the Query phase that we encounter failures. During experimentation, we found out that the error was caused by permission denial errors, which can be attributed towards the number of different accounts we used to query the blockchain. Writes were allowed only when the blockchain permission and Reasoner both allowed a Write operation. The performance evaluation results are shown in Figure 11.

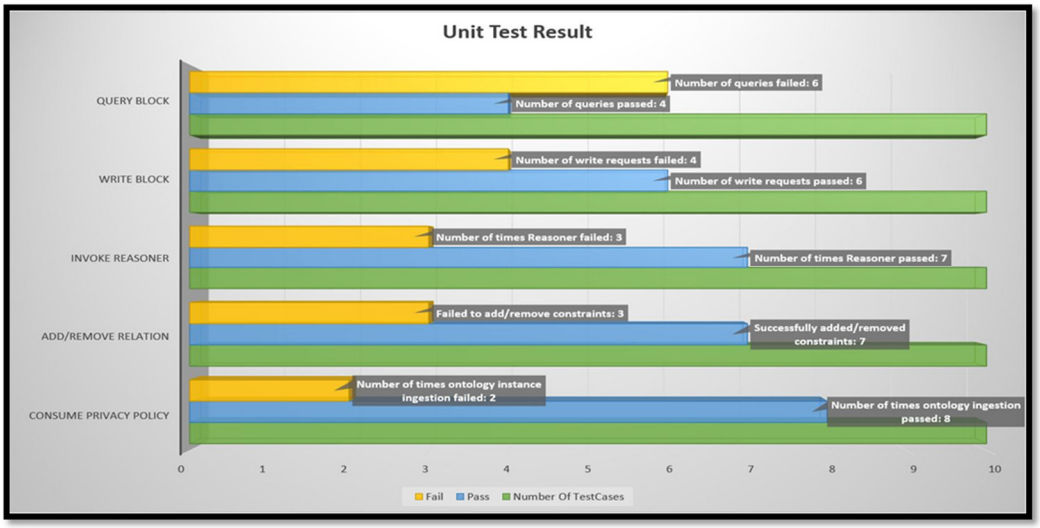


**Figure 11.** Blockchain Sub-System Performance Result.

## 6. Conclusion and Future Work

Sharing of personal data is of vital necessity for service-providers to be able to provide a seamless user experience and to be able to monetize on their service offerings. Unfortunately, such sharing of data to different third-parties and amongst different departments in a single service organization makes the process vulnerable to violation of the privacy policy between the end-user and the service provider. It is required per regulatory guidelines to allow end-users to own and control their data without compromising security or limiting the sharing of the service they have opted for.

Using technologies from the Semantic Web, permissioned Blockchain and NLP/Text Extraction, we have developed a novel methodology to track end-to-end cloud data, including validation of every data operation and recording of data policy provenance. Our architecture enables this by utilizing the blockchain platform as a context-centric access-control model. Based on our architecture, end-users are not required to trust any service-provider or a third-party and are always aware of who is accessing their data and how it will be further used. With a decentralized platform and cloud-based central control, making legal and regulatory decisions about collecting, storing and sharing personal data can now be done automatically using our approach.

Consumers recognize the need to be able to monitor their shared PII information efficiently, with an added impetus on accountability and transparency by the provider. We believe our policy integrated Blockchain based framework can facilitate real time monitoring of all data operations

performed on Cloud datasets. For our purposes, we used a permissioned Hyperledger Fabric blockchain and combined it with robust reasoning capabilities. We focused on integrity, accountability and end-to-end provenance so that the end-system can support secure storage, exchange, and manipulation of PII data and at the same time have accountability. The solution is easy to deploy and use. Moreover, the provenance is based on a machine processable privacy policy ontology, which is populated using the data privacy policy of the service provider.

We continue to improve and enhance our methodology. Some design improvements that we are working on include improving the policy extraction techniques. The relations from the ontology can be discovered by using two approaches. Firstly, a link finding algorithm such as Path Ranking Algorithm can be utilized to find new, possible relations that can exist in the policy. These relations can further be presented to the user. Secondly, this over-head in managing the privacy policies by manually adding relations can be mitigated by migrating the privacy policies as constraints of a Smart Contract. For instance, while the Service Provider accesses personally identifiable information for an end-user, the contract between the concerned parties must include and satisfy the relation *IsDataController*.

Our current implementation accounts for only one consumer and one provider data exchange, and we plan on extending this to include scenarios of one consumer, many providers, and vice versa. Additionally, any privacy policy can thus be looked upon as a smart contract agreement between various stakeholders. Being a single client-single service provider system, issues such as batch query and query frequency are handled keeping the singular source of the query into consideration. We plan to handle query frequency and batch query as part of moving the privacy policy as a smart contract. The proposed model can also be equipped with some necessary data management functions which emphasize further privacy protection: 1. Anonymization: Even before sharing encrypted data, an access management module can anonymize data which removes personally identifiable information if necessary. This will be useful while accessing a set of related data for repurposing (for market basket analysis or other such data manipulations). 2. Communication: Special communication modules can be added to intelligently handle the task of communicating with other related parties for data requests or collaboration. 3. Data backup and recovery of information whenever necessary.

**Author Contributions:** conceptualization, K.P.J.; methodology, K.P.J. and A.B.; software, A.B.; validation, A.B.; writing—original draft preparation, K.P.J. and A.B.; supervision, K.P.J.; project administration, K.P.J.

**Funding:** This research was partially supported by a DoD supplement to the NSF award #1439663: NSF I/UCRC Center for Hybrid Multicore Productivity Research (CHMPR).

**Acknowledgments:** The authors wish to thank Prof. Anupam Joshi and Prof. Tim Finin for their valuable advice to this project.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Joshi, K.P.; Yesha, Y.; Finin, T. Automating Cloud Services Life Cycle through Semantic Technologies. *IEEE Trans. Serv. Comput.* **2014**, *7*, 109–122.
2. Gupta, A.; Mittal, S.; Joshi, K.P.; Pearce, C.; Joshi, A. Streamlining Management of Multiple Cloud Services. In Proceedings of the 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 2–27 July 2016; pp. 481–488.
3. Mittal, S.; Joshi, K.P.; Pearce, C.; Joshi, A. Automatic extraction of metrics from SLAs for cloud service management. In Proceedings of the 2016 IEEE International Conference on Cloud Engineering (IC2E), Berlin, Germany, 4–8 April 2016; pp. 139–142.
4. Joshi, K.; Gupta, A.; Mittal, S.; Pearce, C.; Joshi, A.; Finin, T. Semantic Approach to Automating Management of Big Data Privacy Policies. In Proceedings of the IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016.
5. Banerjee, A.; Joshi K.P. Link before you share: Managing privacy policies through blockchain, 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 4438–4447.

6. Kim, H.; Laskowski, M. *Toward an Ontology-Driven Blockchain Design for Supply-Chain Provenance, Intelligent Systems in Accounting, Finance and Management*; Wiley Online Library: New York, NY, USA, 2018; pp. 18–27.
7. TOVE Ontologies. Available online: http://www.eil.utoronto.ca/theory/enterprise-modelling/tove/ (accessed on 4 February 2019).
8. Zyskind, G.; Nathan, O.; Pentland, A. Decentralizing Privacy: Using Blockchain to Protect Personal Data. In Proceedings of the 2015 IEEE Security and Privacy Workshops, San Jose, CA, USA, 21–22 May 2015; pp. 180–184.
9. Kosba, A.; Miller, A.; Shi, E.; Wen, Z.; Papamanthou, C. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 839–858.
10. Sutton, A.; Samavi, R. Blockchain Enabled Privacy Audit Logs. In *The Semantic Web—ISWC 2017, ISWC 2017, Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2017; Volume 10587.
11. Zhang, N.J.; Todd, C. A privacy agent in context-aware ubiquitous computing environments. In *CMS 2006. LNCS*; Leitold, H., Markatos, E.P., Eds.; Springer: Heidelberg, Germany, 2006; Volume 4237, pp. 196–205.
12. Byun, J.; Li, N. Purpose based access control of complex data for privacy protection. In Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies, Vienna, Austria, 1–3 June 2015; ACM: New York, NY, USA, 2005.
13. de Montjoye, Y.V.; Shmueli, E.; Wang, S.S.; Pentlan, A.S. openPDS: Protecting the privacy of metadata through safeanswers. *PLOS ONE* **2014**, *9*, e98790
14. Chen, L.; Hoang, D.B. Novel Data Protection Model in Healthcare Cloud. In Proceedings of the 2011 IEEE 13th International Conference on High Performance Computing and Communications (HPCC), Banff, AB, Canada, 2–4 September 2011; pp. 550–555.
15. OAuth Protocol. Available online: https://tools.ietf.org/html/rfc6749 (accessed on 4 February 2019.
16. Belaazi, M.; Rahmouni, H.B.; Bouhoula, A. An Ontology Regulating Privacy Oriented Access Controls. In Proceedings of the International Conference on Risks and Security of Internet and Systems (CRiSIS 2015), Mytilene, Greece, 20–22 July 2015.
17. Jansen, W.; Grance, T. NIST SP 800-144 Guidelines on Security and Privacy in Public Cloud Computing. Available online: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-144.pdf (accessed on 4 February 2019).
18. NIST SP 800-53. Available online: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf (accessed on 4 February 2019).
19. NIST Special Publication 800-122, Guide to Protecting the Confidentiality of Personally Identifiable Information (PII). Available online: http://csrc.nist.gov/publications/nistpubs/800-122/sp800-122.pdf (accessed on 4 February 2019).
20. Regulation 2016/679 of the European Parliament. Available online: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679 (accessed on 21 January 2019).
21. European Commission, Protection of Personal Data. Available online: https://ec.europa.eu/info/law/law-topic/data-protection_en (accessed on 4 February 2019).
22. Privacy Alliance. Available online: http://www.privacyalliance.org/resources/ppguidelines/ (accessed on 4 February 2019).
23. Federal Trade Commission (FTC). Available online: https://www.ftc.gov/tips-advice/business-center/privacy-and-security (accessed on 4 February 2019).
24. Beesley C, 7 Considerations for Crafting an Online Privacy Policy, U.S.S.B.A. (United States Small Business Administration). Available online: https://www.sba.gov/blogs/7-considerations-crafting-online-privacy-policy (accessed on 4 February 2019).
25. The Truth about Blockchain, Harvard Business Reviews. Available online: https://hbr.org/2017/01/the-truth-about-blockchain (accessed on 4 February 2019).
26. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Consulted* **2012**, *28*, 2008.
27. Hyperledger Project. Available online: https://www.hyperledger.org/ (accessed on 4 February 2019).
28. Kemme, B.; Alonso, G. A new approach to developing and implementing eager database replication protocols. *ACM Trans. Database Sys.* **2000**, *25*, 333–379.
29. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y. et al. Hyperledger fabric: A distributed operating system for permissioned

blockchains. In Proceedings of the Thirteenth EuroSys Conference (EuroSys '18), Porto, Portugal, 23–26 April 2018.
30. Lassila, O.; Swick, R. *Resource Description Framework (RDF) Model and Syntax Specification*; WWW Consortium. 1999. Available online: https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/ (accessed on 4 February 2019).
31. McGuinness, D.; van Harmelen, F. *OWL Web Ontology Language Overview*; W3C Recommendation, World Wide Web Consortium, 2004. Available online: https://www.w3.org/TR/owl-features/ (accessed on 4 February 2019).
32. Elluri, L.; Nagar, A.; Joshi, K.P. An Integrated Knowledge Graph to Automate GDPR and PCI DSS Compliance. In Proceedings of the IEEE International Conference on Big Data, Seattle, WA, USA, 10–13 December 2018.
33. ACL COLING Dataset. Available online: https://usableprivacy.org/data (accessed on 4 February 2019).
34. Lieberman, J.; Singh, R.; Goad, C. W3C Geospatial Ontologies. Available online: https://www.w3.org/2005/Incubator/geo/XGR-geo-ont/ (accessed on 4 February 2019).
35. Karuna Joshi, Ontology for Data Privacy Policy, Available online: http://ebiquity.umbc.edu/resource/html/id/370/Ontology-for-DataPrivacy-Policy (accessed on 4 February 2019).
36. State Laws related to Internet Privacy. Available online: http://www.ncsl.org/research/telecommunications-and-informationtechnology/state-laws-related-to-internet-privacy.aspx (accessed on 4 February 2019).
37. Financial Industry Business Ontology (FIBO). EDM Council. Available online: https://spec.edmcouncil.org/fibo/ (accessed on 4 February 2019).
38. Finin, T.; Joshi, A.; Kagal, L.; Niu, J.; Sandhu, R.; Winsborough, W.; Thuraisingham, B. ROWLBAC—Representing Role Based Access Control in OWL. In Proceedings of the 13th Symposium on Access Control Models and Technologies, Estes Park, CO, USA, 11–13 June 2008.
39. Sharma, N.K.; Joshi, A. Representing Attribute Based Access Control Policies in OWL. In Proceedings of the ICSC, Laguna Hills, CA, USA, 4–6 February 2016.