Published in proceedings of 6th International IBM Cloud Academy Conference, ICACON 2018, Japan

A Semantically Rich Knowledge Representation of PCI DSS for Cloud Services

Ankur Nagar and Karuna Pande Joshi
Information Systems Department
University of Maryland, Baltimore County
Baltimore, MD 21250
{anku2, karuna.joshi}@umbc.edu

***Abstract*- Organizations often use Cloud based services to manage their financial transactions, like fees payment, invoice payment etc., with their end users. These services allow payment using credit cards and thus need to adhere to the Payment Card Industry Data Security Standard (PCI DSS) standard. To effectively manage this policy, educational institutions dealing with card transaction currently monitor their payment gateway flow manually in order to be compliant. We propose a novel approach to automate this process using semantic web technology and natural language processing. In this paper we describe our technical approach and the ontology that we have developed to automatically extract key terms and rules from various compliance documents and represent them in a machine processable knowledge graph. We used Semantic Web's Web Ontology Language (OWL) to create this knowledge graph which is machine processable and so can contribute significantly in automating the continuous monitoring of credit card PII data operation, transfer and sharing**

***Keywords*- Cloud Computing, Financial Regulations, Semantic Web, Natural Language Processing, PCI DSS policy**

1. INTRODUCTION

Personally Identifiable Information (PII), specifically when it is used in financial transactions, needs to be securely managed by Cloud based service providers. Educational organizations like Universities, Colleges and K-12 School systems often have access to their students' PII details while using cloud-based services. These may include student's financial information like credit card numbers to pay fees online, banking details etc. Since a vast majority of online payment made by students or their guardians to education institutions today is via the credit or debit card, every educational institution needs to protect the misuse of cardholder's personal card data. Thus, one of the key compliance regulation for data protection which every institution using cloud-based service, for e-commerce activity, must adhere to is the Payment Card Industry Data Security Standard (PCI DSS) [4]

PCI DSS is an information security which issues a widely accepted set of policies and procedures intended to optimize the security of credit, and debit card transactions and protect cardholders against misuse of their personal information [4]. The PCI DSS was created jointly in 2004 by four major credit-card companies, viz. Visa, MasterCard, Discover and American Express [4]. The Payment Card Industry Security Standards Council (PCI SSC) was then formed and these companies aligned their individual policies to create the PCI DSS [4]. The PCI DSS has 12 mandatory requirements which are described in the next section.

The PCI DSS regulation is currently consolidated in text documents and so is not machine processable. Hence, it currently requires significant human and time effort to ensure continuous compliance evaluation. We are developing techniques to automatically extract key terms and rules from various compliance documents and represent them in a machine processable knowledge graph [1][2][3].

We have created a semantically rich policy-based knowledge representation of the PCI DSS regulation and present it in this paper. We used Semantic Web's Web Ontology Language (OWL) [6] to create this knowledge graph. This knowledge graph is machine processable and so can contribute significantly in automating the continuous monitoring of credit card PII data operation, transfer and sharing. Section 2 of the paper covers related work in this area. Section 3 describes the

key PCI DSS control requirements. Section 4 describes the methodology we developed using Information Retrieval, Natural Language Processing and Semantic Web technologies for creating the PCI DSS knowledge graph. We conclude with our future work in section 5.

2. RELATED WORK

In the card processing ecosystem vulnerability to financial fraud is high if the organization dealing with these transaction lack securities. PrivacyRights.org states that more than 510 million records with sensitive information have been breached since January 2005 [4]. Merchants, being at the center of card payment flow, need to adherence to technology and standard security procedures to protect theft of cardholder data. This is where the PCI DSS council plays a vital role in protecting cardholder's personal data as it applies to all entities that store, process, and/or transmit cardholder data. It covers technical and operational system components included in or connected to cardholder data [4]. If a merchant accepts or processes payment cards, they must comply with the PCI DSS [4]. Section 3 lists the PCI DSS controls policy in detail.

2.1 Semantic Web

In a Cloud environment, consumers and providers should be able to exchange information, queries, and requests with some assurance that they share a common domain knowledge of interest. This is critical not only for the data but also for the policies followed by service consumers or providers [1]. The interoperability requirement is not just for the data itself, but even for describing services, their service level agreements, quality related measures, and their policies for sharing data [1].

One possible approach to such issue is to implement Semantic Web techniques which will come handy in modeling and reasoning about services related information. We have used this approach for automating cloud privacy policy documents [17] [8]. The Semantic Web deals primarily with data instead of documents. It enables data to be annotated with machine understandable meta-data, allowing the automation of their retrieval and their usage in correct contexts.

Semantic Web technologies include languages such as Resource Description Framework (RDF) and Web Ontology Language (OWL) for defining ontologies and describing meta-data using these ontologies as well as tools for reasoning over these descriptions [1]. 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.

In one of our prior works, we described a new integrated methodology for the lifecycle of IT services delivered on the cloud and demonstrate how it can be used to represent and reason about services and service requirements and so automate service acquisition and consumption from the cloud [1]. In this paper also, we are building a knowledge graph with the help semantic web technology.

3. PCI DSS

This section of our paper will describe the key requirements, which are needed by any institution to be PCI DSS compliant. The goal of the PCI DSS is to protect cardholder data wherever it is processed, stored or transmitted [4]. The security controls and processes required by PCI DSS are vital for protecting cardholder account data, including the PAN – the primary account number printed on the front of a payment card. Merchants and any other service providers involved with payment card processing must never store sensitive authentication data after authorization [4]. This includes sensitive data that is printed on a card or stored on a card's magnetic stripe or chip – and personal identification numbers entered by the cardholder [4]. Figure1 represents details on card of a cardholder.

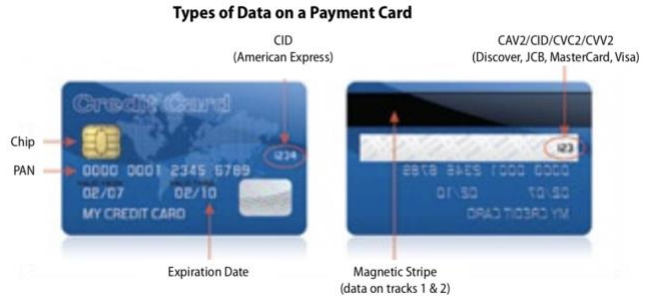

Figure 1: Data on Payment card (Payment Card Industry Security Standards Council, 2016) [4]

In general, if an organization deals with credit card transactions, it must adhere to the security policies listed in the sections below. The PCI Data Security Standard specifies twelve requirements for compliance, organized into six logically related groups [4]. These policies are part of latest PCI DSS Version 3.2 released in 2016 [4]. These 6 groups that hold all the generic 12 requirements are described below.

3.1. Build and maintain a Secure Network

The provider is required to build and maintain a secure communication network. This group of PCI DSS has two compliance rules:

- 'Install and maintain a firewall configuration to protect cardholder data'. The network configuration and its security requirements should be the shared by the IT team and cloud service providers [4].

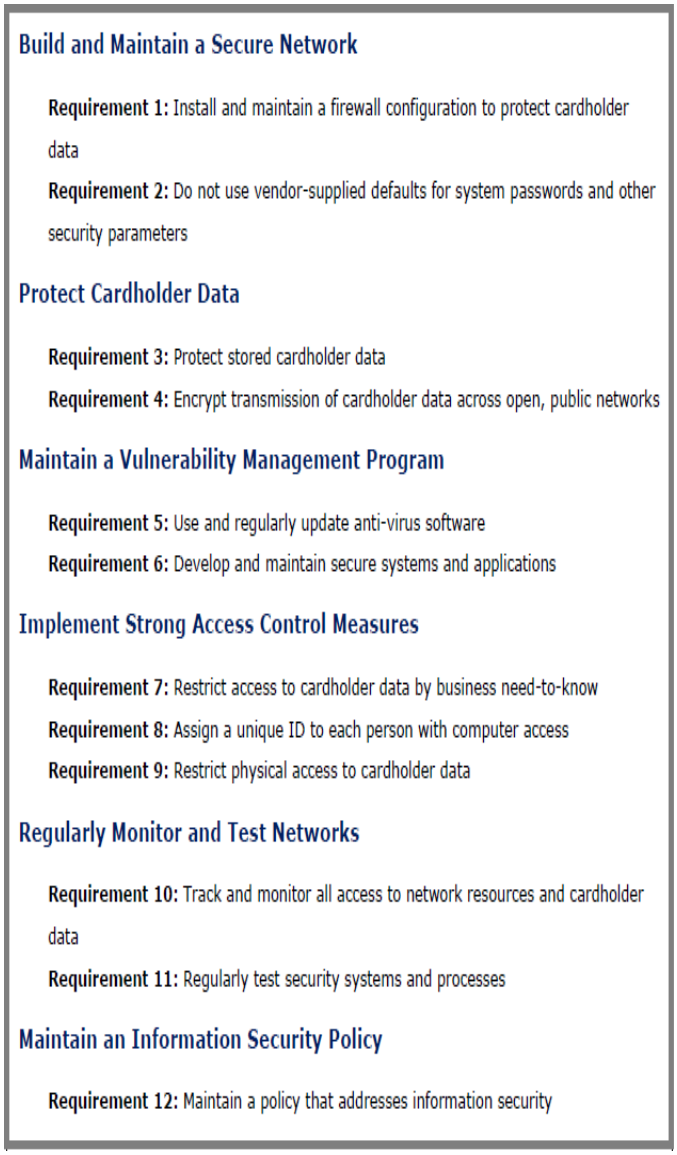

Figure 2: PCI DSS Requirements (Payment Card Industry Security Standards Council, 2016) [4]

- 'Define system password and its security parameters'. This means that all the default passwords supplied by the providers should be changed when a system is getting installed in the configured network [4].

3.2. Protect Cardholder Data

This group regulates merchants to not store any personal card data which may not be needed and whenever the data is transmitted over the internet then it must have encryption data. This class has two compliance properties:

- 'Protect stored cardholder data'. This means that only necessary data should be stored and should purge any unnecessary data at least every quarter. PAN details should be masked, the first six and last four digits are the maximum number of digits you may display [4]. Also, PAN details must be made unreadable wherever it is being stored [4].

- 'Encrypt transmission of cardholder data across open, public networks'. This compliance rule makes sure that strong cryptography and encryption technologies like SL/TLS, SSH or IPSec etc. should be used to safeguard sensitive cardholder data during transmission over any networks [4].

Guidelines for Cardholder Data Elements

		Data Element	Storage Permitted	Render Stored Account Data Unreadable per Requirement 3.4
Account Data	Cardholder Data	Primary Account Number (PAN)	Yes	Yes
		Cardholder Name	Yes	No
		Service Code	Yes	No
		Expiration Date	Yes	No
	Sensitive Authentication Data[1]	Full Magnetic Stripe Data[2]	No	Cannot store per Requirement 3.2
		CAV2/CVC2/CVV2/CID	No	Cannot store per Requirement 3.2
		PIN/PIN Block	No	Cannot store per Requirement 3.2

Figure 3: Cardholder's Data Elements (Payment Card Industry Security Standards Council, 2016) [4]

3.3. Maintain a Vulnerability Management Program

This group talks about to create a program which will help in finding weaknesses in the security protocols, implementation design, cloud network design that may get violated. It includes the following three rules:

- 'Use and regularly update anti-virus software or programs'. All the systems and servers should have anti-virus software's to prevent malicious activity. At the same time, anti-virus services should be running in the background and generating auditing logs [4].

- ‘Develop and maintain secure systems and applications’. This policy ensures that all the patches must be installed on time whenever the patches are generated by the vendors. Any changes to the system components, coding of applications must be done through proper change and control procedures. Also, firewall protection should be ensured for any public facing web applications [4].

3.4. Implement Strong Access Control Measures

The implementation policy should make sure that access to cardholder’s data should be given to only authorized personnel and every person should have a unique ID when they try to access the data. This group includes the below policies:

- ‘Restrict access to cardholder data by business need to know’. This policy ensures that the access is limited to system components and cardholder’s data. Also, an access control protocol for systems components should be in place for multiple users and it must restrict access based on a user’s needs and should be set to “deny all” unless specifically authorized [4].

- ‘Assign a unique ID to each person with computer access’. These policies imply that every person accessing data should have a unique ID for tracing individual’s activity. Also, there should be a two-factor authentication for remotely logging into the network for, such as making use of RSA token or other technologies that facilitate two-factor authentication [4]

- ‘Restrict physical access to cardholder data’. This ensures that proper facility controls should be applied to the cardholder data environment and individual with authorization only be allowed to access cardholder’s data. For visitors, proper token should be given with an expiry and a visitor log must be maintained for tracking purposes [4].

3.5. Regularly Monitor and Test Networks

Networks be it cloud, physical or wireless are the most vulnerable to get compromised, so to be secure during all times, these networks and its parameters should be monitored heavily, and flaws should be tested if found and patched during monitoring. It includes the following two policies:

- ‘Track and monitor all access to network resources and cardholder data’. This ensures that an established process should be implemented to link access of individuals to system components. Log activities of the system components must be reviewed daily, and audit trail history must be retained for at least one year so that three months of activity is available immediately [4].

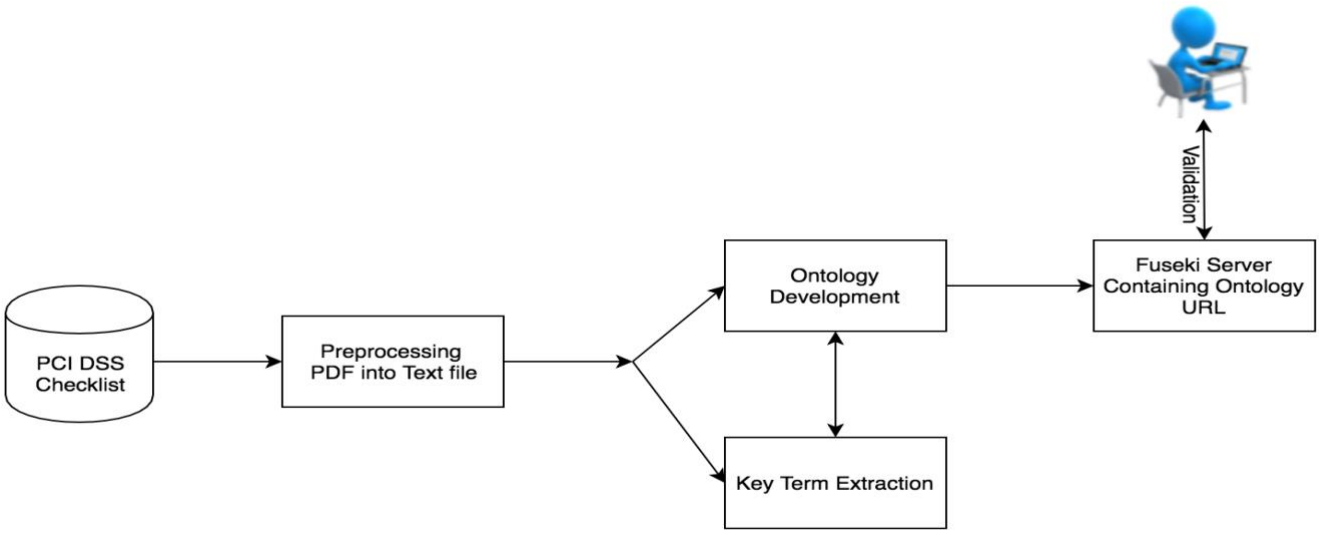

Figure 4: Architecture Flow

- ‘Regularly test security systems and processes’. This ensures that test procedures should be in place to detect access points and unauthorized users. Also, external and internal penetration testing should be performed, including network and application-layer penetration tests at least annually [4].

3.6. Maintain an Information Security Policy

A strong security policy must be in place to inform employees about their respective duties for compliance. This class contains the final requirement of PCI DSS policies:

- ‘Maintain a policy that addresses information security for all personnel’. This ensures that the PCI DSS policies that has been established, published, maintained have descriptive clear definitions of the procedures that everyone in the system knows thoroughly; and such policy must be reviewed at least once a year [4].

4. METHODOLOGY

We have developed a semantically rich methodology to automatically extract key terms from the PCI DSS text and use it to develop a detailed Ontology or Knowledge Graph using the OWL language. Our overall architecture is illustrated in Figure 4. As a first step, we created a repository of the PCI DSS checklist [4]. As this document was only available as a PDF document, we converted it to a text format using Python code and its libraries. We next extracted key terms from the preprocessed file and built our knowledge graph. We next uploaded our knowledge graph to an Apache web server.

Our framework mainly consists of these three stages:

- **Extracting key terms:** We extracted high frequency key terms from 12 defined PCI DSS rules (see section 3 for the rules). Using techniques of Natural language processing, we were able to automatically extract the key terms from the text document the detailed explanation is listed in section 4.1.

- **Ontology Development:** We have defined a detailed ontology for PCI DSS compliance policy using the OWL language and is detailed in section 4.2. For creating the knowledge graph or Ontology of the PCI DSS standard, we utilized the Protégé tool [17].

- **Validation:** We validated the design of our knowledge graph against the publicly available PCI DSS polices of Universities and firms. In section 4.3 we have described this process.

4.1 Extracting Key terms

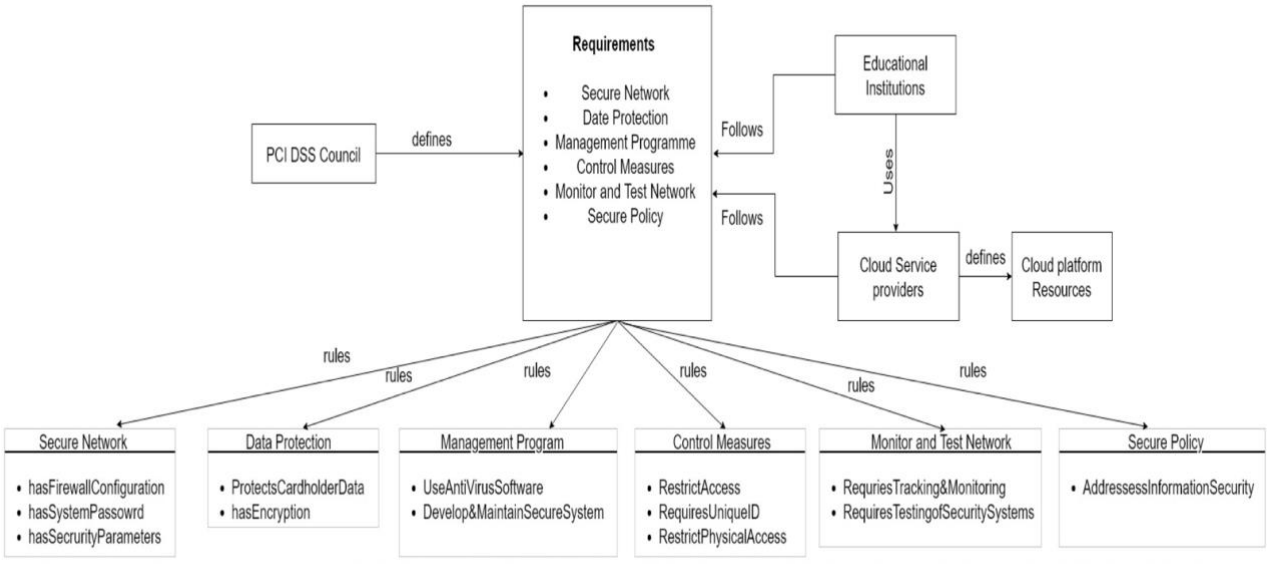

Figure 5: High Level Ontology of PCI DSS

To build our PCI DSS ontology, we had to identify the key concepts embedded in the document. Hence, first stage of our methodology is to identify the key terms and sections of the document. As described in section 3, we identified the key 6 groups in PCI DSS policy that contain the compliance requirements. These were identified as key classes to be captured by our Ontology (section 4.2).
To identify the key terms and rules in the PCI DSS document checklist, we next reviewed the main 12 rules that have to be adhered to by organizations dealing with transaction of credit or debit card data. We extracted the key terms from these rules by identifying the most frequently occurring words in the PCI DSS policy to ensure unbiased term extraction. Since the PCI DSS checklist is very long, while extracting relevant key words from the document we removed some of the generic words like stop words. We filtered out the stop words before processing rest of the document.

Some common stop word functions including words like 'must', 'should', 'should not', 'must not' etc., were retained by us as these words are part of policies, defining the logic and phrase behind it. Using Python, we developed code for extracting relevant terms from the pre-defined set of PCI DSS compliance rules. The most frequently occurring and relevant terms which that were identified include 'ensure' 'security' 'control' 'access' 'unauthorized'. Their frequency in the document is listed below in Table 1.

Key terms	Frequency
Maintain	10
Control	13
Establish	5
Access	43
unauthorized	6
Ensure	10

Table 1: Key terms

4.2 Ontology / Knowledge graph

An ontology can be defined as a common vocabulary for researchers which will be helpful in sharing information of a domain [13]. For our framework, we have used OWL and RDF languages to capture the rules defined by the PCI DSS Council. These are open source languages developed by WWW Consortium (W3C) and so our ontology, which is in public domain, can be easily adopted by organizations. It is also platform independent and so can be easily integrated with Cloud based services.

RDF [6] is a language which is helpful in encoding knowledge on web space so as to make the information understandable to electronic agents searching for domain related information. In general, ontologies are used to capture information for domain of interest. In this knowledge graph of PCI DSS, we have also incorporated our previous Cloud Services SLA (Service Level agreement) ontology [11] [8] to include classes pertaining to Cloud Service Providers (CSPs).

Our knowledge base consists of six different class which incorporate the 12 requirements. Figure 5 illustrates our ontology. The main stakeholder entities are PCI DSS Council, Educational Institutions and Cloud Service Providers. In our ontology we have six classes having two or more subclasses in it. Each class are disjoint from other classes which means that an individual (or object) cannot be an instance of more than one of these six classes [17]. Each of the sub classes has their own properties. Some of the properties that we have developed using PCI DSS checklist [4] are as mentioned in Table 2 below.

PCI DSS Requirement	Property
Requirement 8	assign_unique_user_name
Requirement 8	authenticate_users
Requirement 1	change_vendor_supplied_devices_and_passwords
Requirement 11	Deploy_file_integrity_monitoring_tools
Requirement 5	Deploy_updated_anti-virus_software
Requirement 2	develop_configuration_standards
Requirement 12	develop_daily_operational_security_procedures
Requirement 3	Don't_store_sensitive_authentication_data

Table 2: Classes Properties (Payment Card Industry Security Standards Council, 2016) [4]

4.3 Validating Knowledge Graph

In this section we briefly describe the process of validating our PCI DSS ontology. To validate our PCI DSS knowledge graph, we populated it with publicly available policy documents. We have hosted this ontology on an Apache Jenna Fuseki server [11] [12] and executed simple SPARQL queries to retrieve rules from these documents. Table 3 lists the organizations whose Cloud policy documents we referenced to validate our ontology. We were able to include a wide variety of organizations in our evaluation including Universities, Private entities and Cloud service providers. On all of the documents we were looking for PCI DSS polices to create our dataset which helped us in validating our ontology and make it machine

Institutions	Policies
Ultracart	https://www.ultracart.com/resources/pci-compliance.html
Ebay	https://www.ebay.com/gds/PCI-Compliance-FAQ-/10000000009055977/g.htmlhttps://www.ebay.com/gds/PCI-Compliance-FAQ-/10000000009055977/g.htm
Coastal University	https://www.coastal.edu/policies/pdf/univ-its%20480%20pci%20dss-%20september%202016.pdf
Boston University	https://www.bu.edu/cfo/comptroller/departments/cashier/resources/pci-data-security-standards/
Shopify	https://www.shopify.com/legal/terms

Table 3: List of policies used in Validation

processable so to contribute significantly in automating the continuous monitoring of credit card PII data operation, transfer and sharing.

5. CONCLUSION & FUTURE WORK

Currently the PCI DSS compliance policy is managed as a text file. As a result, a lot of manual effort is required to make sure, the institution is marked as compliant. We are working to automate the process using Semantic web technologies, like OWL and RDF. In this paper we have described the PCI DSS knowledge graph developed by us. This knowledge graph is machine processable and so can contribute significantly in automating the continuous monitoring of credit card PII data operation, transfer and sharing.

The main goal of our system is to create a Knowledge Base of all facts present in PCI DSS document. We envision a Knowledge Base with PCI DSS term definitions and policy rules that can be used as a checklist by cloud service providers operating in the domain. The end user will have access to this Knowledge Base and will be able to query it using a SPARQL interface.

As part of our future work, we will also explore the security measures that exist for Mobile and other Device applications dealing in payment transactions data.

6. REFERENCES

[1] Karuna P Joshi, Aditi Gupta, Sudip Mittal, Claudia Pearce, Anupam Joshi, and Tim Finin. Semantic Approach to Automating Management of Big Data Privacy Policies. In Proceedings, IEEE BigData, 2016.

[2] Srishty Saha, Karuna P. Joshi, Renee Frank, Michael Aebig, Jiayong Lin. Automated Knowledge Extraction from the Federal Acquisition Regulations System (FARS). InProceedings, IEEE International

[3] Karuna Pande Joshi et al., "Automating Cloud Services Lifecycle through Semantic technologies", Article, IEEE Transactions on Service Computing, January 2014,

[4] Payment Card Industry (PCI) Data Security Standard, Version 3.2 April 2016
https://www.pcisecuritystandards.org/document_library

[5] Suman Ramkhelawan, Baby Gobin-Rahimbux, Zarine Cadersaib. PCI-DSS Requirements in the Mauritian Hospitality Industry. In proceedings, IEEE International Conference EmergiTech, 2016

[6] D. McGuinness, F. Van Harmelen, et al., OWL web ontology language overview, W3C recommendation, World Wide Web Consortium, 2004.

[7] Natalya F, D. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology, 2004

[8] Ontology for Cloud Services SLA (Service Level Agreement), Karuna Pande Joshi and Tim Finin https://ebiquity.umbc.edu/resource/html/id/344/Ontology-for-Cloud-Services-SLA-Service-Level-Agreement

[9] S. Mittal, K. P. Joshi, C. Pearce, and A. Joshi, "Automatic Extraction of Metrics from SLAs for Cloud Service Management", InProceedings, 2016 IEEE International Conference on Cloud Engineering (IC2E 2016), 2016

[10] Apache Jena Fuseki Server,
https://jena.apache.org/documentation/fuseki2/

[11] Getting started with RDF SPARQL queries and inference using Apache Jena Fuseki, Christine Draper, https://christinemdraper.wordpress.com/2017/04/09/getting-started-with-rdf-sparql-jena-fuseki/

[12] OWL: Web Ontology Language, Sean Bechhofer,
https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_1073

[13] S. Soderland, Learning to extract text-based information from the world wide web, in KDD, vol. 97, 1997, pp. 251254

[14] Understanding the 12 Requirements of PCI DSS - Practical steps to achieve and maintain compliance, Opinion Piece [Online]. www.dimensiondata.com/globalpresence. [Accessed: 21- Feb- 2016]

[15] Meeting PCI DSS When Using a Cloud Service Provide - ISACA , http://www.isaca.org

[16] K. P. Joshi and C. Pearce, "Automating Cloud Service Level Agreements Using Semantic Technologies," 2015 IEEE International Conference on Cloud Engineering, Tempe, AZ,2015, pp. 416-421, doi: 10.1109/IC2E.2015.63

[17] Protégé Editor- Protégé Tool http://protege.stanford.edu.