

**MODELING AND ANALYSIS OF SECURITY STANDARDS FOR WEB
SERVICES AND CLOUD COMPUTING**

by

Ola Ajaj

A Dissertation Submitted to the Faculty of
the College of Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Florida Atlantic University

Boca Raton, FL

December 2013

Copyright by Ola Ajaj 2013

MODELING AND ANALYSIS OF SECURITY STANDARDS FOR WEB SERVICES
AND CLOUD COMPUTING

by

Ola Ajaj

This dissertation was prepared under the direction of the candidate's dissertation advisor, Dr. Eduardo B. Fernandez, Department of Computer & Electrical Engineering and Computer Science, and has been approved by the members of his supervisory committee. It was submitted to the faculty of the College of Engineering and Computer Science and was accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

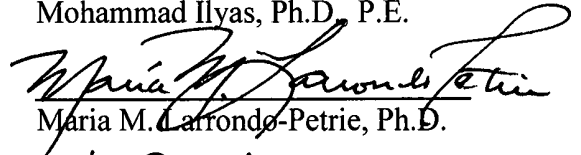
SUPERVISORY COMMITTEE:



Eduardo B. Fernandez, Ph.D., P.E.
Dissertation Advisor



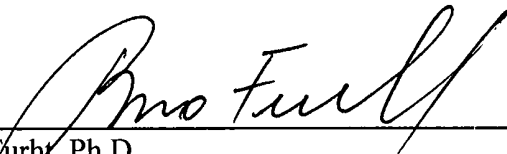
Mohammad Ilyas, Ph.D., P.E.



Maria M. Larrondo-Petrie, Ph.D.



Mihaela Cardei, Ph.D.



Borko Furht, Ph.D.
Chair, Department of Engineering & Electrical
Engineering and Computer Science



Mohammad Ilyas, Ph.D., P.E.
Interim Dean, College of Engineering and Computer Science



Barry T. Ross, Ph.D.
Dean, Graduate College

August 14, 2013
Date

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Eduardo B. Fernandez, for his guidance during these years. He has been a true mentor by supporting me during my research in. I would also like to express my gratitude to my committee members, Dr. Mohammad Ilyas, Dr. Maria Petrie, Dr. Mihaela Cardei, , and the members of the Secure Systems Research Group for all their advice and constructive comments of this dissertation. I dedicate this degree and I wish to express deep appreciation to my family and friends for their continuous support and encouragement throughout this Ph.D.'s program; without which I wouldn't make it and so I am unconditionally appreciative.

ABSTRACT

Author: Ola Ajaj
Title: Modeling and Analysis of Security Standards for Web Services and Cloud Computing
Institution: Florida Atlantic University
Dissertation Advisor: Dr. Eduardo B. Fernandez
Degree: Doctor of Philosophy
Year: 2013

Cloud Computing is a new computing model consists of a large pool of hardware and software resources on remote datacenters that are accessed through the Internet. Cloud Computing faces significant obstacles to its acceptance, such as security, virtualization, and lack of standardization. For Cloud standards, there is a long debate about their role, and more demands for Cloud standards are put on the table. The Cloud standardization landscape is so ambiguous. To model and analyze security standards for Cloud Computing and web services, we have surveyed Cloud standards focusing more on the standards for security, and we classified them by groups of interests. Cloud Computing leverages a number of technologies such as: Web 2.0, virtualization, and Service Oriented Architecture (SOA). SOA uses web services to facilitate the creation of SOA systems by adopting different technologies despite their differences in formats and protocols. Several committees such as W3C and OASIS are developing standards for web services; their standards are rather complex and verbose. We have expressed web

services security standards as patterns to make it easy for designers and users to understand their key points. We have written two patterns for two web services standards; WS-SecureConversation, and WS-Federation. This completed an earlier work we have done on web services standards. We showed relationships between web services security standards and used them to solve major Cloud security issues, such as, authorization and access control, trust, and identity management. Close to web services, we investigated Business Process Execution Language (BPEL), and we addressed security considerations in BPEL and how to enforce them. To see how Cloud vendors look at web services standards, we took Amazon Web Services (AWS) as a case-study. By reviewing AWS documentations, web services security standards are barely mentioned. We highlighted some areas where web services security standards could solve some AWS limitations, and improve AWS security process. Finally, we studied the security guidance of two major Cloud-developing organizations, CSA and NIST. Both missed the quality of attributes offered by web services security standards. We expanded their work and added benefits of adopting web services security standards in securing the Cloud.

MODELING AND ANALYSIS OF SECURITY STANDARDS FOR WEB SERVICES AND CLOUD COMPUTING

TABLES	xii
FIGURES.....	xiii
1. INTRODUCTION.....	1
2. A PATTERN FOR THE WS-SECURECONVERSATION STANDARD OF WEB SERVICE.....	11
2.1. Web Services Standards	11
2.2. A Pattern for the WS-SecureConversation Standard of Web Services.....	14
2.2.1. Introduction.....	14
2.2.2. Intent	16
2.2.3. Context.....	16
2.2.4. Problem.....	16
2.2.5. Solution	18
2.2.6. Implementation	24
2.2.7. Example Resolved	25
2.2.8. Known uses.....	26
2.2.9. Consequences.....	26
2.2.10. Related Patterns	28
2.2.11. Conclusion	28
3. A PATTERN FOR THE WS-FEDERATION STANDARD FOR WEB SERVICES.....	30
3.1. Introduction	31
3.2. A pattern for WS- Federation standard for web services	34
3.2.1. Intent	34
3.2.2. Example	34
3.2.3. Context.....	35
3.2.4. Problem.....	35

3.2.5.	Solution	37
3.2.6.	Implementation	39
3.2.7.	Implementation	46
3.2.8.	Example Resolved	48
3.2.9.	Known uses	48
3.2.10.	Consequences.....	49
3.2.11.	Related Patterns	50
3.2.12.	Conclusion	51
4.	RELATING WEB SERVICES SECURITY STANDARDS.....	52
4.1.	Relationships between Web Services Security Standards.....	53
4.1.1.	A pattern for WS-Policy	54
4.1.2.	A pattern for WS-Trust	56
4.1.3.	A pattern for WS-SecureConversation	57
4.1.4.	A pattern for WS-Federation	58
4.2.	The relationship between WS-policy and WS-Trust.....	59
4.3.	The relationships between WS-policy, WS-Trust and WS-SecureConversation.....	61
4.4.	Relationships between WS-policy, WS-Trust, WS-SecureConversation and WS-Federation.....	63
5.	ADDING SECURITY TO BPEL WORKFLOWS OF WEB SERVICES	65
5.1.	Introduction	65
5.2.	Background	68
5.3.	An example for a collaborative business process.....	70
5.4.	Threats to the activities	74
5.5.	Stopping or mitigating the threats	77
5.6.	Conclusions and Future Work.....	80
6.	THE NEED FOR CLOUD STANDARDS	82
6.1.	The Concept of a Standard	82
6.1.1.	De jure Standards.....	82
6.1.2.	De facto Standards	83
6.1.3.	Consortium Standards	84
6.2.	Aspects of a Good Standard	85
6.3.	The Need for Cloud Computing Standards	86

6.3.1.	Facilitate Communications	87
6.3.2.	Security	88
6.3.3.	Ranks.....	90
6.3.4.	Users Awareness.....	90
6.3.5.	Comparing Providers	91
6.4.	Summary	91
7.	SURVEY OF SECURITY STANDARDS FOR CLOUD COMPUTING	93
7.1.	Importance of Cloud Computing Security Standards	93
7.2.	Issues of Cloud Computing Standardization.....	97
7.2.1.	Security	98
7.2.2.	Virtualization	99
7.2.3.	The Lack of Standardization.....	99
7.2.4.	Lock-In Problem	101
7.3.	Cloud Standardization Efforts.....	102
7.3.1.	Distributed Management Task Force (DMTF).....	105
7.3.2.	Institute of Electrical and Electronics Engineers (IEEE).....	106
7.3.3.	National Institute of Science and Technology (NIST)	107
7.3.4.	Open Group.....	108
7.3.5.	Open Grid Forum.....	108
7.3.6.	OpenStack Foundation.....	109
7.3.7.	Organization for the Advancement of Structured Information Standards (OASIS)	109
7.3.8.	Storage Networking Industry Association (SNIA).....	110
7.4.	Survey of Security Standards for Cloud Computing.....	111
7.4.1.	Problem with NIST Categorization of Cloud Computing Standards.....	111
7.4.2.	Discussion	136
7.5.	Patterns for security standards of Cloud Computing	136
7.6.	Summary	139
8.	CASE STUDY: SECURITY STANDARDS IN AMAZON WEB SERVICES	141
8.1.	AWS SOAP-based web services Vs. REST-based web services.....	142
8.2.	AWS Security Best Practices	143
8.3.	Security Standards in AWS.....	143
8.4.	Evaluating Security Aspects in AWS.....	147

8.4.1.	Shared Responsibility Model.....	148
8.4.2.	Customer Awareness	148
8.4.3.	Implementation	150
8.4.4.	Protection against Attacks.....	151
8.4.5.	Operating Systems Protection.....	151
8.5.	Web Services Security Standards in AWS.....	152
8.6.	Security Flaws in Amazon Web Services	153
8.7.	Summary	156
9.	CONTRIBUTION OF WEB SERVICES SECURITY STANDARDS FOR SECURING THE CLOUD.....	159
9.1.	Matching Web Services Standards with Cloud Key Security Guidance	159
9.1.1.	CSA-Security Guidance for Critical Areas of Focus in Cloud Computing	160
9.1.2.	CSA-The Notorious Nine: Cloud Computing Top Threats	162
9.1.3.	NIST- Guidelines on Security and Privacy in Public Cloud Computing (SP 800-144).....	163
9.1.4.	NIST-Cloud Computing Synopsis and Recommendations.....	164
9.1.5.	Cloud Computing Quality Attributes.....	165
9.2.	Summary	167
10.	ADOPTING WEB SERVICES SECURITY STANDARDS IN CLOUD COMPUTING.....	168
10.1.	Issue: Need for Policies.....	169
10.1.1.	Example	170
10.1.2.	Solution: WS-Policy.	170
10.1.3.	Example Resolved	172
10.2.	Issue: Authorization and Access Control	173
10.2.1.	Example	173
10.2.2.	Solution: XACML	174
10.2.3.	Example Resolved	175
10.3.	Issue: Data Protection during Transmission.....	176
10.3.1.	Example	177
10.3.2.	Solution: WS-Security.	177
10.3.3.	Example Resolved	178
10.4.	Issue: Trust	179

10.4.1. Example	182
10.4.2. Solution: WS-Trust	182
10.4.3. Example Resolved	184
10.5. Issue: ID Management	185
10.5.1. Example	186
10.5.2. Solution: WS-Federation	186
10.5.3. Example Resolved	189
10.6. Summary	189
11. CONCLUSIONS AND FUTURE WORK.....	190
12. REFERENCES	197

TABLES

Table 1: Mitigating or stopping threats to a business process using security policies	77
Table 2: Cloud Computing Standardization Efforts	103
Table 3: Survey of Security Standards for Cloud Computing	114
Table 4: List of patterns for security standards of Cloud Computing.....	138
Table 5: An overview of some AWS services and the corresponding SOAP or REST APIs.....	142
Table 6: Matching Web Services with CSA-Critical Areas of Focus in Cloud Computing.....	161
Table 7: Matching Web Services with CSA Top Threats in Cloud Computing.....	162
Table 8: Matching Web Services Security Standards with NIST- Guidelines on. Security and Privacy	163
Table 9: Matching Web Services Security Standards with NIST-Cloud Computing Synopsis and Recommendations.....	164
Table 10: Matching Web Services Standards with Cloud Computing Quality Attributes.....	166

FIGURES

Figure 1: Class Diagram for the WS-SecureConversation Pattern	19
Figure 2: Sequence Diagram for establishing a SCT to create a context.....	22
Figure 3: Sequence Diagram amending an existing SCT	23
Figure 4: Pattern diagram for web services security standards.....	32
Figure 5: Component diagram for the WS-Federation Pattern	39
Figure 6: Class diagram for the WS-Federation Pattern	43
Figure 7: Sequence Diagram Obtain access to a resource using identity token	45
Figure 8: Sequence Diagram accessing a resource using a pseudonym ID	46
Figure 9: Pattern diagram for web services security standards.....	53
Figure 10: Class diagram for the WS-Policy Pattern.....	55
Figure 11: Class diagram for the WS-Trust Pattern.....	57
Figure 12: Relationship between WS-Policy and WS-Trust	60
Figure 13: Relationships between WS-Policy, WS-Trust and WS-SecureConversation	62
Figure 14: Relationships between WS-Policy, WS-Trust, WS-SecureConversation and WS-Federation	64
Figure 15: An example of a collaborative business process	70
Figure 16: A BPEL Activity diagram for reserving a room	73
Figure 17: A BPEL sequence diagram for reserving a room.....	74
Figure 18: Some threats to the BPEL Activity diagram for reserving a room	76
Figure 19: Addressing security considerations for BPEL.....	80

1. INTRODUCTION

As described by NIST, Cloud Computing is “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. [Hog11].

Even though there are many advantages to adopting Cloud Computing, there are also some significant obstacles to its acceptance. One important issue is security, followed by virtualization, lack of standardization and Cloud standards adoption.

Despite the large amount of active work in developing standards for the Cloud, there is currently a long debate about the role of standards in the Cloud. On one side parties who see the Cloud as a completely new approach that needs a completely new set of standards, and on the other side parties who see the Cloud as a technology built based on existing technologies that already have standards. In one part of this thesis we balanced both parties' points of view, defined what is a standard from our perspective, and why do we need standards in Cloud Computing.

Even though standards have not been a requirement for the vast growth of the Cloud, more and more demands for Cloud standards are put on the table. The Cloud

standardization landscape is so ambiguous because there isn't a central body or forum to control the process of standardization, despite the efforts made many people and organizations on that direction. NIST did a good job of listing the standards relevant to Cloud Computing [Nis13a], but their categorization of Cloud Computing standards is like one size fits all, it is not clear, and it is hard for users and researchers to use.

To clear the picture of Cloud standards, we surveyed general Cloud standards, we focused more on the standards for security, and we classified them by groups of interests. This should make exploring Cloud standards easier for both users and researchers.

Cloud Computing leverages a number of computing models and technologies; such as Service Oriented Architecture (SOA), Web 2.0, virtualization and other Internet-based technologies. This led us to the next point where we explained the relations between Cloud, SOA and web services.

We can describe SOA (Service Oriented Architecture) as a loosely coupled set of software services and functions for the purpose of supporting business functions. Those software services and technologies can be assembled and reassembled according to the business requirement. It is an architectural style useful to implement business functions. SOA is mainly applied to business information systems using web services standards and technologies, and is rapidly becoming a standard approach for enterprise IT systems.

SOA applies to Cloud Computing at many levels. Generally speaking, Cloud Computing derives its association to SOA by advantage of the characteristic of loose-coupling. This is true since loose-coupling is a key concern within SOA, mostly due to

the desire to be agnostic to the underlying technology, topology, lifecycle and organization on which any service is implemented in an information system.

Moving forward to Cloud Computing and we can see the same objective growing from a business perspective, i.e., being able to create and control services and utilities to solve customers' needs, regardless of the location and technology on which those services are implemented.

Adding to this, business people are more interested of thinking what services they are offering to their customers and partners not it in the IT or SOA sense of service, but in terms of their business value to the markets. Similarly, they tend to think of the services they consume from business partners as services. To give an example, one company may provide travel services, another company provides financial services, and a third one provides HR services. Leveraging all those business services through the Internet and Cloud Computing is an evolutionary target for all those partners to expand their markets and attracts more customers.

SOA was never dependent on one specific technology (including web services) for its success. However, web services standards do facilitate the creation of SOA systems and, therefore, adoption of technology despite their differences in formats and protocols. This led us to study web services and how they could contribute to the growth of SOA and Cloud.

Web services intend to provide an application integration technology that can be used over the Internet in a secure, interoperable and trusted manner. Web services in the

Cloud are very closely related to SaaS, but instead of delivering complete applications over the internet, the service providers provide APIs that enable customers to use functionality over the Internet. Some examples are Google Maps, ADP payroll processing, and credit card processing services. The definition of Web APIs is: “an API is typically a defined set of Hypertext Transfer Protocol request messages along with a definition of the structure of response messages, usually expressed in an Extensible Markup Language or JavaScript Object Notation format” [Wik13].

How Cloud Computing and web services are related? Well, one important thing that has distinguished WS-* related technologies is that they are standards and have raised a strong community of support, and this has enabled higher qualities of service. But when you need to go beyond that, the web services standards are particularly useful for solving Cloud Computing issues, such as asynchronous messaging, and metadata exchange, security, policies, trust, and federation of identities. In one chapter of this thesis, we used web services security standards to solve major Cloud issues, such as, need for policies, authorization and access control, data protection during transmission, trust, and ID management.

Several committees such as W3C and OASIS are developing web services standards. Their standards are rather complex and verbose and it is not easy for designers and users to understand their key points. By expressing web services security mechanisms and standards as patterns, we can verify if an existing product implementing a given security mechanism supports some specific standard [Fer06c].

In this thesis we wrote two patterns for two web services standards; WS-SecureConversation, and WS-Federation. Both could be used to solve two major challenges of Cloud which are exchange numerous amount of messages and ID management. This work completed an earlier work we have done before on patterns for web services standards found here [Aja10a, Aja10b].

WS-SecureConversation defines mechanisms for establishing and sharing Security Context Token (SCTs), and deriving keys from security contexts to authenticate messages between parties [Wss07]. The WS-Federation standard defines a framework with additional federation mechanisms that extend these specifications [Wsf09]. Later on the following chapters, we will use those two web services security standards beside other WS-* standards to solve major Cloud issues. All the patterns follow the POSA [Bus06] template and are similar to the style of the security patterns written in [Sch06].

Writing patterns for web services standards for the purpose of writing patterns is not enough, we need to show how they are related to offer a complete solution that is independent, reusable, and flexible. In this thesis we draw a pattern diagram for web services security standards, and explained what degree of dependency a pattern could have with other patterns. We defined better relationships between patterns for web services security standards.

In a related subject to web services, we investigated Business Process Execution Language (BPEL). As a web service composition language, BPEL can be a convenient and effective means for application integration over the Internet in typical Business-to-

Business (B2B) interaction scenarios [Wsb07]. However, for BPEL to keep its promises it is necessary to provide more support for security. Vendors and companies are looking to have their requirements of authentication, integrity and confidentiality satisfied. In this thesis we addressed security considerations in BPEL and how to enforce them.

Cloud Computing is a large pool of resources; that we can dynamically manage to scale up and down to match the load, where users pay per usage. The resources include hardware and systems software on remote datacenters that are accessed through the Internet. Key benefits of adopting Cloud Computing are elasticity, multi-tenancy, resource utilization and pay-per-usage. These new features provide what it needs to leverage large infrastructures through virtualization or resource management, but these large pools of resources are not necessarily located in the same country (same physical location) nor even on the same continent, which make it hard to keep track of what resources are used and where. Compliance with data-handling regulations is also difficult to fulfill. In this scenario, auditing is a challengeable task too, since those resources are susceptible to volatility and power outages.

Not only these new benefits make it hard or let's say impossible to reuse traditional security, privacy and trust mechanisms in the Cloud, but also, they raise significant issues and concerns that need to be fully addressed.

While Cloud adoption is expected to speed up in the coming years, companies are still cautious about the Cloud as the right delivery medium for their services. The dominant concern is security. Since Cloud Computing is a relatively new Computing

model, where there is a great deal of hesitation about how to achieve security at all levels (e.g., host, virtualization, and data) [Ros12]. The main question whether data is safe in the Cloud, and how they can offer an on-demand service while preserving industry compliance. Enterprises are right to hesitate about investing heavily into the Cloud without some guarantees about protection. The detailed nature of the Cloud is vague and open for attackers, and the virtual nature of the Cloud makes protecting on-demand environments a complex process. Other issues were regarding compliance, privacy and legal matters [Kpm10].

Security concerns are spotted in different areas such as external data storage, dependency on the open “public” internet, lack of user control, multi-tenancy and integration with internal security. The current form of traditional security mechanisms such as authorization, authentication, trust, privacy, and ID management are still valid for Clouds [Liw09]. We are going to look at some of those Cloud issues and try to solve them using web services security standards.

In this work, we make the following contributions:

1. We present a pattern for the WS-SecureConversation (Chapter 2) that described how a web service can authenticate requester messages, how requesters can authenticate services, and how to establish mutually authenticated security contexts.
2. We present a pattern for the WS-Federation (Chapter 3) that describes how to manage and broker the trust relationships in a heterogeneous federated

environment, including support for federated identities, sharing of attributes, and management of pseudonyms.

3. We define what relationships of web services security standards (Chapter 4) exist between web services security standards. We draw a complete pattern diagram, and then we show what degree of dependency a pattern has with other patterns in details.
4. Adding Security to BPEL workflow of web services (Chapter 5). We have presented an approach that enumerates the threats to a given BPEL process. We consider UML activity diagrams for collaborative business processes and show how to list the possible threats and attacks that could happen in order to define the appropriate and suitable countermeasures to stop or mitigate them.
5. The need for Cloud Computing standards (Chapter 6) aims to solve the confusion about the need for standards in Cloud Computing, which is either urgent, nonexistent, or some wherein between. We start this chapter by trying to define what a standard is, and then we explain what makes a good standard. We end by listing the main factors of why we need standards for Cloud Computing.
6. Survey of security standards for Cloud Computing (Chapter 7). Numerous standards from different organizations are available in the Cloud market. We survey here work on security standards for Cloud Computing and we classify them in groups depending on their functionalities. We also include standards that although not developed for Cloud Computing, have an impact on the use of

clouds. The Cloud standardization landscape is so ambiguous because there isn't a central body or forum to control the process of standardization. We list the main issues of Cloud Computing standardization. We briefly present some industry efforts.

7. In chapter 8, we take Amazon Web Services (AWS) as a case-study, since Amazon is one of the major Cloud vendors. We intend to see how Cloud vendor look at web services standards, and whether they take them into consideration while offering services. While reviewing AWS documentations, we have noticed Amazon barely mentioning web services security standards as solutions to be considered. We highlight some areas where web services standards could solve some AWS limitations. We identify other spots where AWS can improve its security process by adopting or encouraging users to use web services security standards.
8. Chapter 9 aims to select two major Cloud-developing organizations, which are CSA and NIST, and examine their security guidance of securing the Cloud. We notice both missed the quality of attributes offered by web services standards. We expand their work and added one more dimension of web services security standards. Our addition adds benefits and advantages of adopting web services security standards in securing Cloud Computing.
9. Chapter 10 matches major Cloud security issues with solutions offered by web services security standards, some of which include: message exchange, transport,

security, reliability, trust, federation of identities. Web services security standards give better solution to the security problem so that Cloud Computing are easier to be accept in the business cases require high protection of the data and information.

This thesis is organized as a collection of papers which implies some overlap and repetition of topics. It includes the following chapters: Chapter 2 presents a pattern for the WS-SecureConversation standard of web services. Chapter 3 is for a pattern for the WS-Federation standard for web services. In chapter 4, we present pattern diagram for web services standards. Chapter 5 presents adding Security to BPEL Workflows of Web Services. Chapter 6 describes the need for Cloud Computing standards. Chapter 7 surveys security standards for Cloud Computing, investigates issues for Cloud Computing standardization and proposes solutions using web services security standards. Chapter 8 takes Amazon Web Services (AWS) as a case study and evaluates its security standards, and how web services security standards can add to AWS. Chapter 9 talks about how web services security standards contribute into securing the Cloud. Finally, adopting web services standards in Cloud Computing is the goal for chapter 10. Finally, Chapter 11 is about conclusion and future work.

2. A PATTERN FOR THE WS-SECURECONVERSATION STANDARD OF WEB SERVICE

In this thesis we wrote two patterns for two web services standards; WS-SecureConversation, and WS-Federation. Both could be used later to solve two major challenges of Cloud which are exchange numerous amount of messages and ID management. This work completed an earlier work we have done before on patterns for web services standards found here [Aja10a, Aja10b].

2.1. Web Services Standards

Over time, different languages, mechanisms, and tools have been developed on different software and hardware platforms for specifying and implementing a variety of security mechanisms, such as encryption and access control. In a web service setting, security mechanisms protect the confidentiality and integrity of the data in transit, and the data at rest. Furthermore, protection of the information must not only consider simple two way client-server interactions, but also extend to more complex interactions, as in the case of business process implemented through multiple web services. The need for providing end-to-end security through distributed and heterogeneous security mechanisms called for the development of standards for web services security, with the ultimate goal of making interoperable different implementations of the same security functions.

Web services technology is being used industry wide to implement interoperable service oriented architectures (SOAs). This technology contains a set of evolving related standards that aims to address SOA goals and challenges. Organizations looking to lower the cost of development and maintenance for their systems, while staying more flexible in terms of capabilities, consider web services standards as a possible solution. A big reason behind adopting Web services standards is their key quality attributes such as interoperability, extensibility, and modifiability [Obr07].

Many organizations are working to create open standards, but the three key organizations are: The Organization for the Advancement of Structured Information Standards (OASIS) whose job is to create the infrastructure and implementation of Web services standards. World Wide Web Consortium (W3C), in charge of (HTTP) XML, SOAP, and other standards. W3C contains many committees whose goals are to build and maintain Web standards (usually described as “recommendations”). The Web Services Interoperability Organization (WS-I) provides practical guidance, best practices, and resources for developing interoperable Web services solutions.

Web services standards have a significant number of technological companies including Microsoft, IBM, Oracle, BEA and others. Those companies are actively participating on creating web services standards, have fully support them, and have create software components built on interoperable standards, and integrate those components into products. Ultimately, the goal of using web services standards is to build a system by installing products developed by different companies and to allow those products to work together seamlessly.

One of the goals of Web services standards is to support interoperable machine-to-machine interaction over a network. This is accomplished by using XML-based messaging technologies such as: Web Services Description Language (WSDL), the Simple Object Access Protocol (SOAP), and the Universal Description, Discovery, and Integration (UDDI). These, beside any additional new standards, are managed by various standards bodies and entities. The web services security standards originally foreseen by the IBM and Microsoft framework [Ibm02]. [Inn07] published a web services standards overview that has 60 standards classified in 12 categories. [Fer10] did a complete survey of web services security in terms of standards and industrial practice.

Web services standards once implemented offer a complete secure solution that in short is:

- *Independent*: from the underlying execution technology and application platforms.
- *Extensible*: to address new requirements and/or exploit new security technologies.
- *Reusable*: Web Services built using web services standards are easy to reuse as appropriate in other services.
- *Flexible*: Can accommodate existing heterogeneous mechanisms that is, different encryption algorithms, different access control mechanisms, and so on.
- *Composable*: Support for composite applications such as business process flows.

Several committees such as W3C and OASIS are developing web services standards. Their standards are rather complex and verbose and it is not easy for designers

and users to understand their key points. By expressing web services security mechanisms and standards as patterns, we can verify if an existing product implementing a given security mechanism supports some specific standard [Fer06c].

2.2. A Pattern for the WS-SecureConversation Standard of Web Services

Abstract:

When using web services, the involved parties need to address two main concerns: first, each party needs to determine if it can trust the credentials of the other party; and second, how to protect their data and how to provide a secure session between the parties. WS-Trust takes care of the first challenge by defining how to establish trust between interacting parties, while WS-Security is in charge of the second part by providing message integrity, confidentiality, and authentication. However, web services exchange multiple messages that increase the overhead of key establishment and decrease performance, which eventually affects business interactions. By defining a shared context among the communicating parties for the lifetime of a communications session that combines secure communications and trusted relationship, we facilitate the process of communication and increase overall performance. This shared context is implemented by the WS-SecureConversation standard, and we present here a pattern for it.

2.2.1. Introduction

Web services interact with users and other web services to conduct business. Those business interactions have different degree of complexity and validity depending on their nature. Some interactions exchange a large number of messages, which adds

complexity and overhead to the message exchanges and thus increase cost. In addition, sometimes users and web services are not predefined and known to each other and a trust relationship must be established before any interaction can happen between them. Those users also might have different requirements and their own policy rules, implementing different security constraints. Addressing all these concerns in one abstract and practical solution will facilitate the interaction between web services. This is the motivation behind the WS-SecureConversation standard.

The Web Services Secure Conversation Standard (WS-SecureConversation) is built on top of the WS-Security, WS-Trust, and WS-Policy standards to provide secure interaction and data exchange between web services [Ibm02]. WS-Security [Wss04] describes how to embed existing security mechanisms such as XML Encryption, XML Digital Signature, and Security Tokens into SOAP messages to provide message confidentiality, integrity, authentication, and non-repudiation. WS-Trust [Wst07] is a standard to support the establishment of trust relationships between web services. WS-Policy [Wsp07] provides specifications that describe the capabilities and constraints of the security (and other business) policies on intermediaries (for example, required security tokens, supported encryption algorithms, and privacy rules) and how to associate policies with services and end points.

To perform its functions, WS-SecureConversation [Wsc09] defines mechanisms for establishing and sharing Security Context Token (SCTs), and deriving keys from security contexts to authenticate messages between parties. This shared context defines who, how, what and for how long each user is able to conduct business. The involved

parties share these SCTs throughout the lifetime of the communication channel [Wsc09], [Ibm02].

2.2.2. Intent

This pattern describes a standard to allow security context establishment and use through the lifetime of the communication session between web services. This security context is used to provide secure communication between web services by extending the mechanisms of WS-Security, WS-Trust, and WS-Policy.

2.2.3. Context

Distributed applications using web services that need to collaborate with each other to perform business workflows using insecure networks, e.g. the Internet.

2.2.4. Problem

Before initiating a conversation with people, we need to know with whom we are talking and what do we want to talk about. We initiate conversations with different persons, based on our roles and interests. The same is true for web services.

The functions of WS-Security which include integrity, confidentiality and authentication of messages are useful for simple or one-way messages; but this solution is impractical and could cause a problem in case of the necessity of parties to exchange multiple messages [Wss04]. If there are multiple message exchanges between service provider and consumer, then the overhead of XML signature and XML encryption are significant.

Further, establishing security relationships is fundamental for the interoperation of distributed systems. Applying relevant policies is needed to make it clear for the users what is allowed or which conditions apply to the use of web services. Without applying relevant policies and trust relationships between the involved parties, web services have no means to assure security and interoperability in their integration and may lose their ability to provide service.

The possible solution to the problem of establishing a secure context is constrained by the following forces:

- ***Securing Context Tokens:*** While communicating using context tokens, web services exchange multiple messages containing sensitive data; we need to provide message protection for this exchange.
- ***Time Restrictions:*** Any interactions or means of communications between web services may be restricted in time. We should be able to amend, renew, or cancel those interactions properly, as needed.
- ***Policy:*** A web service uses policy(ies) to define all the required conditions and constraints that should be met before using that web service. We should reference this policy for verification and proper use.
- ***Overhead:*** Web services exchange multiple messages that add complexity, increase the overhead of key establishment, and decrease performance; we need to keep overhead at a minimum.

- **Interoperability:** Web services and requesters should interact seamlessly despite differences in domains and platforms.

2.2.5. Solution

We define explicitly an artifact that uses a Security Context Token (SCT). The SCT defines what kinds of assertions are required to be satisfied by any interaction between the involved web services and encapsulates the claims and information sent by the requester in order to obtain the required SCT. Once initiated, this SCT can be used to conduct secure communications. All entities involved share a key that has been agreed in order to establish a communication session with their target partners.

Structure

Figure 1 describes the structure of this pattern. Pink classes describe a logical web service connection; blue classes describe the token management structure, while yellow classes describe security tokens and claims.

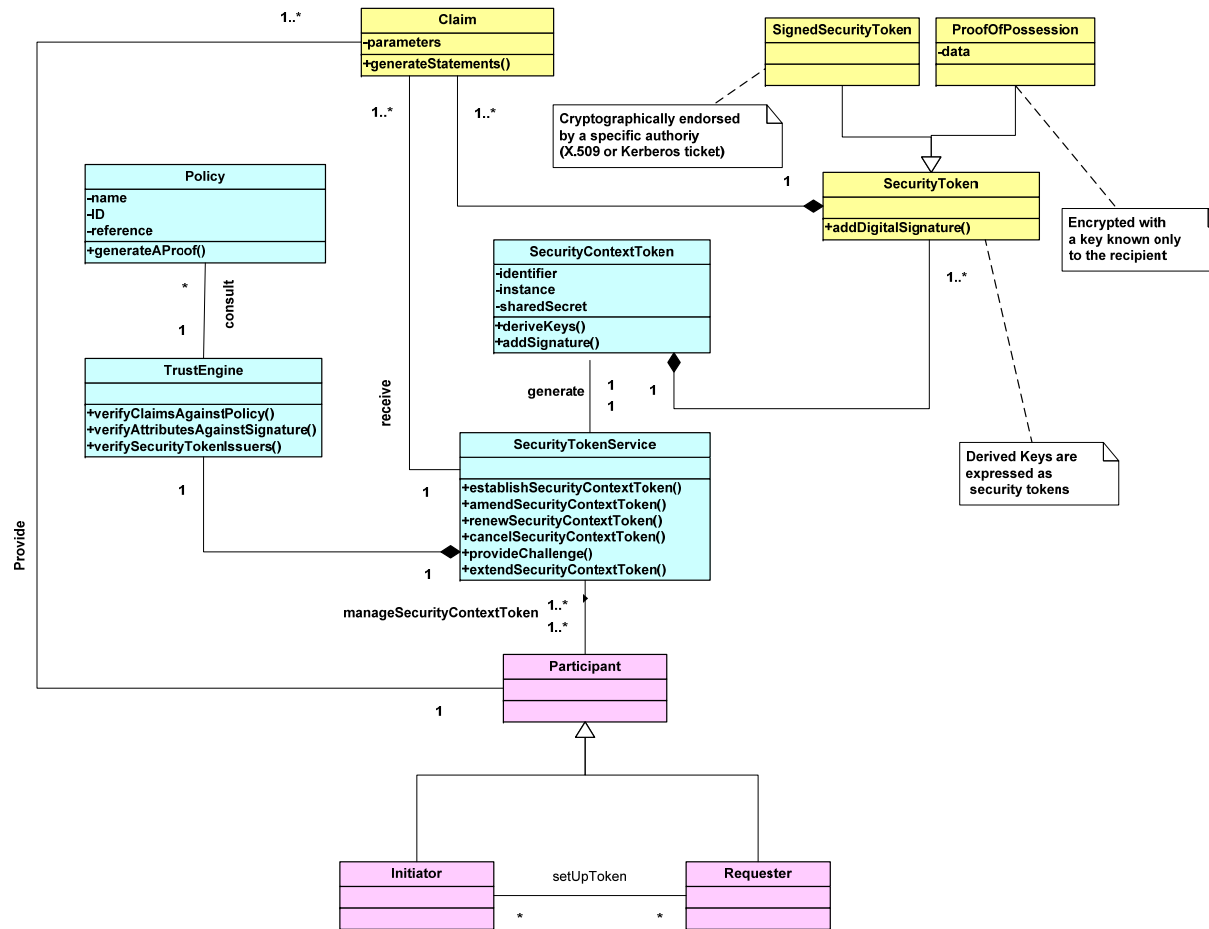


Figure 1: Class Diagram for the WS-SecureConversation Pattern

Participant represents an entity (e.g., human, computer, message, an endpoint, interaction, resource) that is in charge of managing SCTs. It can have the role of: **Initiator**, the one creates a SCT or asks a STS to create one for her, or **Requester**, who asks for a SCT to conduct business and use web services.

A **Claim** is a statement about a client, service or other resource (e.g. name, identity, key, group, privilege, capability, etc.). Claims are assertions such as “I am Adnan”, “I am an authenticated user and I am authorized to print.” A **Security Token** is a collection of claims (such as X.509 certificate, Kerberos ticket, and username).It is responsible for adding signatures to tokens. Security Token also is a generalization of: **Signed Security Token**, that is cryptographically endorsed by a specific authority (e.g. an X.509 certificate or a Kerberos ticket), and **Proof-of-Possession (PoP) Token** which contains a secret *data* parameter that can be used to prove authorized use of an associated security token. Usually, the proof-of-possession information is encrypted with a key known only to the recipient of the POP token.

A **Security Context Token** is a representation of a security context, which in turns refers to an established authentication state with negotiated keys that may have additional security-related properties. Requestors can use SCTs to sign and/or encrypt a series of SOAP messages, known as a conversation, between a message sender and the target web service.

Security Token Service (STS) is a web service that issues SCTs by itself, or relies on another STS to do so using its own trust statement. It produces assertions based

on evidence that it trusts, provides challenge for requesters to ensure message freshness (the message has not been replayed and is currently active), verifies authorized use of a security token and establishes, extends trust among a domain of services. Each **STS** has a **Trust Engine** that evaluates the security-related aspects of a message using security mechanisms and implies a policy to verify the requester's assertions. The **Trust Engine** is responsible for verifying security tokens and verifying claims against policies. A **Policy** is a collection of policy assertions that have their own name, references, and ID.

Dynamics

We describe the dynamic aspects of the WS-SecureConversation using sequence diagrams for the use cases “*Establish a SCT to create a context*” and “*Amend a SCT*”

Establish a SCT to create a context (Figure 2):

Summary: STS creates a SCT using the claims provided by the initiator.

Actors: Initiator, requester.

Precondition: The STS has the required policy to verify the requester claims and the requester provides parameters in form of *claims* and *RequestType* signed by a *signature*.

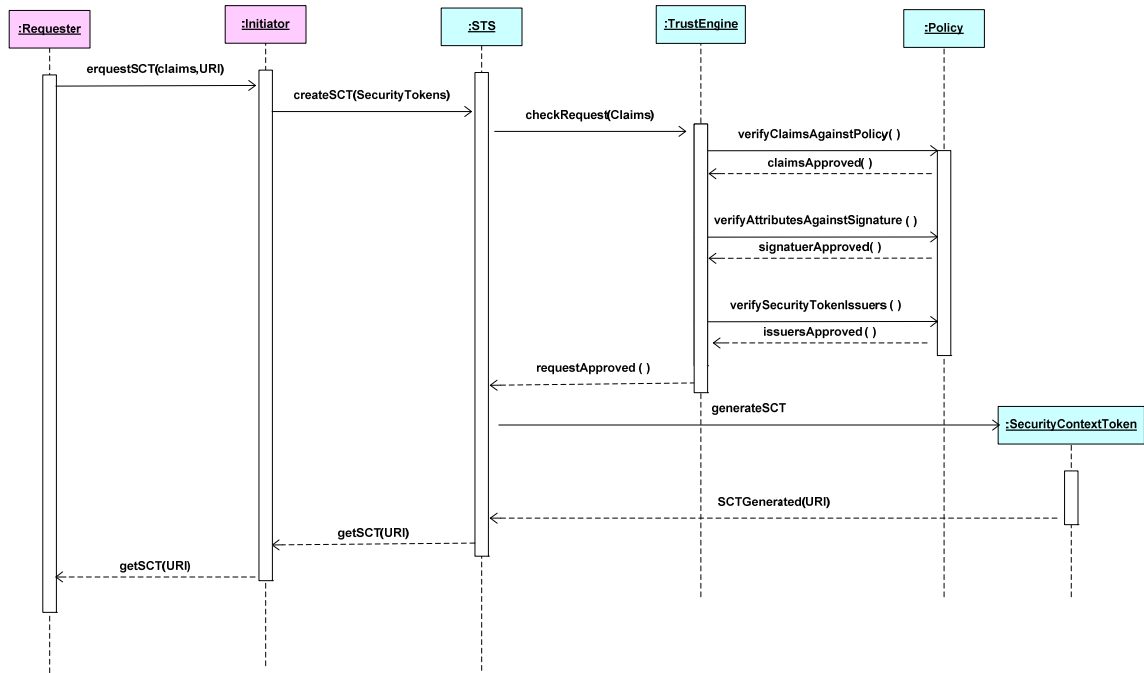


Figure 2: Sequence Diagram for establishing a SCT to create a context

Description:

- The requester asks for a SCT to use a web service.
- The initiator requests a SCT by sending the required parameters of *claims* signed by a *Signature* to the STS in forms of security tokens.
- The STS uses WS-Trust mechanisms of Trust Engine and WS-Policy to check the initiator's claims.
- Once approved, the STS creates a new SCT in the form of an URI and sends it back to the initiator, who in turns sends it back to the requester.

Post condition: The initiator has a SCT that can be used to communicate with other web services.

Amend a SCT (Figure 3):

Summary: A STS will amend an existing SCT to carry additional claims upon the initiator's request.

Actors: Initiator.

Precondition: The initiator owns a SCT.

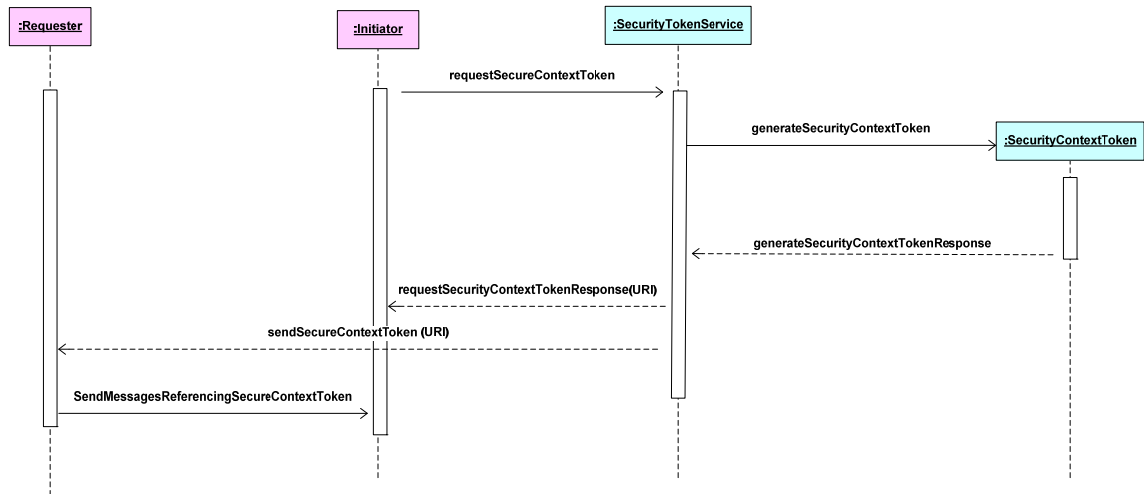


Figure 3: Sequence Diagram amending an existing SCT

Description:

- a. The initiator asks to change a SCT she owned to carry additional SCTs.
- b. The STS asks for the key associated with the SCT as a proof of possession.

- c. The initiator replies back with a signature.
- d. Once approved, the STS adds those additional SCTs to the SCT to form a new SCT and sends it back to the initiator.

Postcondition: The initiator has an amended SCT that can be used to communicate with other web services.

2.2.6. Implementation

When it comes to encrypt a message, SCTs use a symmetric key rather than an asymmetric key, which makes them faster and more efficient. A STC allows a context to be named by a URI, also for reference purposes. The participating web services determine a shared secret to use as the basis of key generation. In other words, shared secrets are in form of derived keys. The derived keys are expressed as security tokens.

In order to assure effective implementation, we need to take in consideration the following:

- Using a service requires a signature to prove knowledge of a security token or set of security tokens. Three possible scenarios are presented to establish a SCT; by one of the communicating parties, by a negotiation/exchange process between the participants, or by a separate STS.
- Although the messages exchanged between the involved entities are protected by WS-Security; still three possible issues related to security tokens are handled: security token format incompatibility, security token

trust and namespace differences. The WS-Trust pattern can address these issues by: Defining a request/response protocol (in which the client sends *RequestSecurityToken* and receives *RequestSecurityTokenResponse* and introducing a Security Token Service (STS).

This pattern could be used to guide the development of a product. Users can use this pattern to ask for certain security mechanisms that fit their business goals. Developers and product vendors should be aware that no complete security solution for web services is guaranteed through WS-SecureConversation by itself. WS-SecureConversation should be used in conjunction with other web services standards such as WS-Security, WS-Trust, and WS-Policy for an optimal solution [Wst07]. Implementing various security mechanisms through those standards will lead to an optimal solution. Implementations of those WS standards are beyond the goals of this pattern and were covered in [Has09], [Aja10a], and [Aja10b].

2.2.7. Example Resolved

Ajiad now has the ability to automate the business relationships with its partners by assuming that all partners are registered and by issuing customers unique IDs. In this case, *Ajiad* provides an intermediate link between the customers and its partners and plays the role of negotiator as well as third-party player who is looking to satisfy both sides. *Ajiad* now can offer a SCT Service for its business partners, who may find useful ways to take advantage of credit processing and other of its services, *Ajiad* now has new business opportunities.

2.2.8. Known uses

- WS-SecureConversation is used in the Microsoft Web Services Enhancement 2.0 toolkit [Gud04].
- WS-SecureConversation support in Apache's CXF builds upon the WS-SecurityPolicy implementation to handle the SecureConversationToken policy assertions [Ap13].
- Java applications support in IBM products includes support for WS-SecureConversation [Sos10].
- SAP is using the security context primarily to allow WS-ReliableMessaging to reuse a security context, so that the server can contact the client [Sap13].

2.2.9. Consequences

The WS-SecureConversation pattern presents the following advantages:

- Policy providers now can use mechanisms provided by other web services specifications such as WS-Security, XML Digital Signature [Xds08], and WS-Metadata Exchange [Wme09] to protect the messages needed to create a secure context.
- ***Time restriction:*** We can specify time constraints in the parameters of SCTs, which can specify how long that token is active. Upon expiration, the SCT's holder may amend, renew, or cancel it.

- ***Policy:*** We can implement WS-Policy to support trusted partners by expressing and exchanging their statements of trust expressed as a trust policy.
- ***Overhead:*** For the communication channels that require end-to-end security and have frequent message exchanges, the WS-SecureConversation may reduce the overhead. Using either encryption or signing is better than using both, since combining both produces significantly lower performance [Liu05].
- ***Interoperability:*** STS satisfies the capabilities and constraints of the security (and other business) policies on intermediaries which at the end increase the interoperability between web services. By implementing STS, the WS-SecureConversation framework will be more comprehensive and can carry out secure conversations between parties in different trust domains.

e WS-SecureConversation pattern presents the following liabilities:

- The WS-SecureConversation standard is a lengthy document with a lot of details that were left out to avoid making the pattern too complex. Somebody interested in addressing more details can check the WS-SecureConversation Standard web page.
- No complete security solution for web services is guaranteed through WS-SecureConversation by itself. WS-Secure Conversation should be used in conjunction with other web services standards such as WS-Security, WS-Trust, and WS-Policy for an optimal solution.

2.2.10. Related Patterns

- A Pattern for WS-Security [Has09] defines how to secure SOAP messages applying XML security standards such as XML Encryption and XML Signature.
- A pattern for the WS-Policy standard [Aja10b] describes how to express requirements that are needed or supported by a web service. For instance, it can indicate that a specific signature algorithm must be used when signing a document.
- A pattern for the WS-Trust standard of web services [Aja10a] provides a framework for requesting and issuing security tokens, and to broker trust relationships. It uses WS-Security to transfer the required security tokens, using XML Signature and Encryption to provide confidentiality. This standard may use WS-Policy to specify which security tokens are required at the target.

2.2.11. Conclusion

We presented a pattern for the WS-SecureConversation that describes how a web service can authenticate requester messages, how requesters can authenticate services, and how to establish mutually authenticated security contexts.

Message authentication is useful for simple or one-way messages; parties intending to exchange multiple messages can create a secure session. A security context

is shared among the communicating parties for the lifetime of a communications session. These security context-token-issuance services build on WS-Security, WS-Trust and WS-Policy to transfer the requisite security tokens in a manner that ensures the integrity and confidentiality of those tokens.

3. A PATTERN FOR THE WS-FEDERATION STANDARD FOR WEB SERVICES

Abstract:

The growth in business-to-business commerce, increased mobility and the importance of persistent interactions between involved parties are some of the current industry challenges. To meet these challenges, companies are extending internal systems to external users of different categories (employees, customers and partners). A variety of users who need to interact with a variety of autonomous systems requires a careful handling of identities. Building secured and trust-based relationships among users might require sharing their identity information. Trust relationships should allow identity and policy data to be exchanged between parties independent of platform, application or infrastructure, and avoid redundant work. A Federation describes the technology and mechanisms necessary to systemize this interconnection, and to allow different domains to use identities from different domains. We present here a pattern for the WS-Federation standard. The WS-Federation standard is built on top of the WS-Security, WS-Trust, and WS-Policy standards to define a framework with additional federation mechanisms that extend these specifications.

3.1. Introduction

Web services are a distributed application architecture based on industry standards such as SOAP, XML, WSDL and UDDI. The idea behind implementing web services is to deliver complete and interoperable business solutions for the enterprise. Organizations need a consistent and secure way of expressing what type of credentials and requests they accept, the policies by which they conduct business, and what services are presented to their customers and partners.

Despite the high degree of interoperability between involved parties, and the fact that each individual continues to manage his own user's identities, users still have the choice of sharing and accepting credentials and identities from members from outside their domain. For that reason, the WS-Federation standard defines mechanisms to allow different security domains (realms) to federate their identities, such that authorized access to resources managed in one domain can be provided to principals whose identities are managed in other domains. Those federation mechanisms enable the decision of federating IDs to be based on the declaration (or brokering) of identity, attribute, authentication and authorization assertions between domains. Addressing all these concerns in one abstract solution will facilitate the interaction between web services.

Figure 4 shows a pattern diagram describing the relationships between the patterns for some web services standards. The diagram shows dependencies between the patterns; for example, WS-Security uses policies defined by WS-Policy. Our group has written all these patterns [Fer12a]; this being the last in the group.

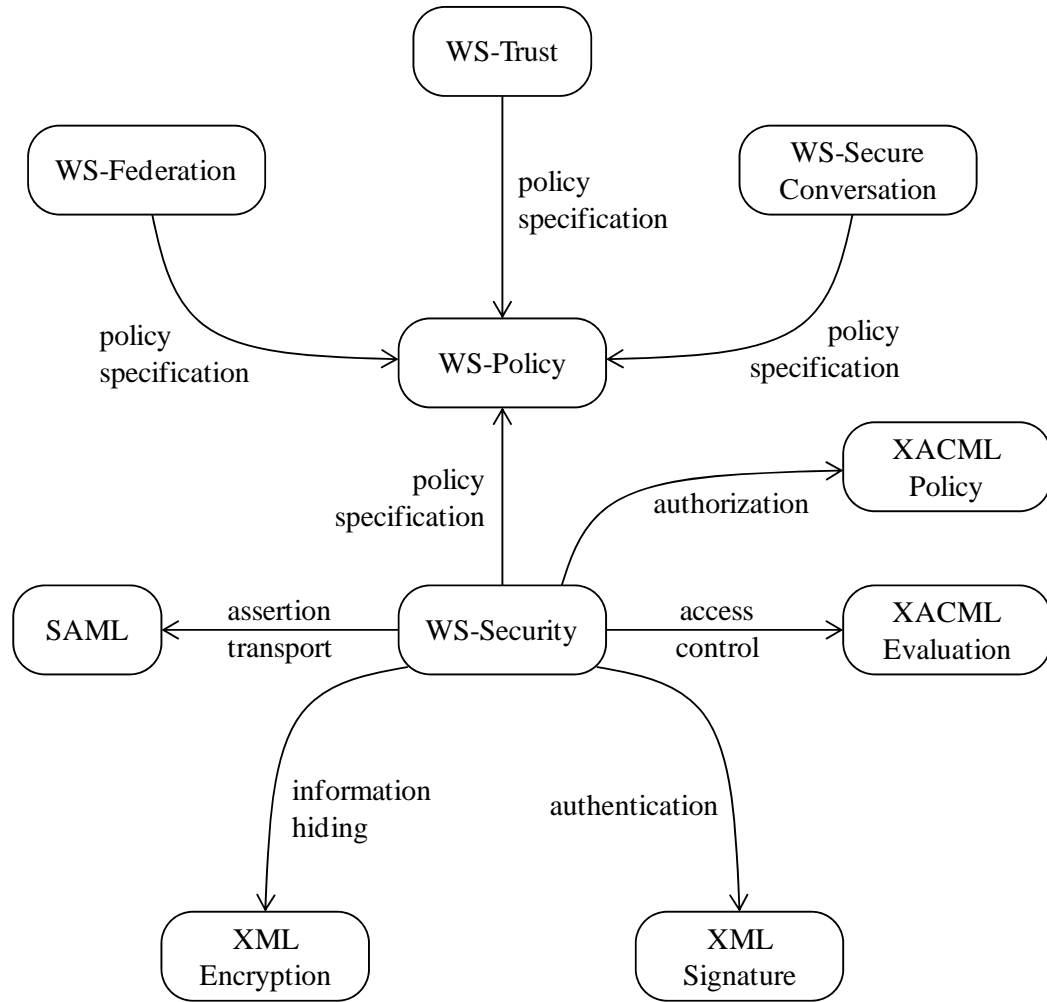


Figure 4: Pattern diagram for web services security standards

The WS-Federation standard is built on top of the WS-Security, WS-Trust, and WS-Policy standards to define a framework with additional federation mechanisms that extend these specifications [Wsf09]. WS-Security [Wss04] describes how to embed existing security mechanisms such as XML Encryption, XML Digital Signature, and Security Tokens into SOAP messages to provide message confidentiality, integrity, authentication, and non-repudiation. WS-Trust [Wst09] is a standard to support the establishment of trust relationships between web services. WS-Policy [Wsp07] provides

specifications that describe the capabilities and constraints of the security (and other business) policies on intermediaries (for example, required security tokens, supported encryption algorithms, and privacy rules) and how to associate policies with services and end points.

Web services standards are rather complex and verbose and it is not easy for designers and users to understand their key points. Our approach is to express web services security mechanisms and standards as patterns. In this way we can verify if an existing product implementing a given security mechanism supports some specific standard [Fer06c]. Inversely, a product vendor can use the standards to guide the development of a product. By expressing standards as patterns, we can compare them and understand them better. For example, we can discover overlapping and inconsistent aspects between them. A standard defines a generic architecture and this is a basic feature of any pattern; it can then be confirmed as a best practice by looking at products that implement the standard (and implicitly the pattern).

Section 2 shows a pattern that describes the WS-Federation standard while Section 3 ends the paper with some conclusions. Our description is intended for users and designers of business workflow systems that use web services. It can also have value for designers of systems implementing this standard. Our audience includes architects and designers of applications using web services and possibly designers of products that use web services.

3.2. A pattern for WS- Federation standard for web services

3.2.1. Intent

Describe a standard to securely share a principal's identity information across trust boundaries, by having it brokered by identity providers and security token issuers without the need for re-authenticating users.

3.2.2. Example

A travel agency implements several business portals to offer services for tickets, hotel and car rental to its customers. Our travel agency is a part of a travel agencies consortium. The goal of this consortium is to expand each partner's business and give it privileges to reach other members domains. Each domain's employees have authorized access to other partners' resources. Each member controls its own resources and has the final access control decision.

Without a well-defined structure of sharing identities with other parties, the travel agency will not be able to determine which travel services to invoke for a given customer, or determine how to allow businesses to directly provide services for customers registered at other (partner) businesses, or allow disparate security domains to broker information on identities, identity attribute and authentication. Not having this structure may lead to losing a valuable business goal of offering integrated travel services.

3.2.3. Context

Distributed applications in a business network, using web services and exchanging messages, need to leverage identity management, and enable cross-domain interactions between partners to pursue business integration goals.

3.2.4. Problem

When it comes to handle identity management in business cross-domain networks, there are two aspects to consider. First: business networks include many partners, systems, applications and business processes, each of which separately controls identity information about its users. Second: within a single company, there is a variety of types of authentications managed independently within the business units. Those types of authentications add more complexity in terms of processing time and smooth interactions between parties.

Cross-domain networks not only face difficulties of allowing customers, partners and end-users to navigate easily between web sites supporting these services without constantly authenticating or identifying themselves to the various sites (unless specified by the underlying policies), but also face challenges in managing access by external users associated with their business partners.

From a company's perspective, more business interactions are better. Leveraging the identities they retain is a must; for which they need trust mechanisms to allow entities to be federated across the collaborating domains, which is difficult to obtain. For those

companies, managing identity increases the risk of damaging their reputation if they release or use information in ways which contradict individual privacy rights.

From an individual's perspective, multiple identities for multiple systems exist, both personal and professional. Individuals navigating between those systems need somehow to have the right credentials for the right service. Having multiple identities for an individual affects providing convenient service for users, and obstructs interactions between business providers.

The possible solution to the problem of sharing and leveraging principals' identities within a federation is constrained by the following forces:

- ***Identity mapping:*** Users may have multiple IDs accessing different accounts in multiple domains. Users should be spared from repeatedly providing their IDs within a federation (unless required by policies).
- ***Identity decentralization:*** ID centralization in business networks results in high costs of identity management in terms of processing time, human resources, and administrative duties. We need to off-load and simplify identity management costs and reduce duplication of efforts.
- ***Degree of Security:*** Even though different parties participating within a federation might have different security architectures and different security policies, those parties should not alter neither their architectures nor their policies in order to comply with other partners' constraints.

- ***Interoperability:*** Institutions can interoperate across organizational and technical boundaries (i.e., various operating systems or security platforms). We need to have common IDs for providers to interoperate.
- ***Privacy practices of institutions:*** Data exchanged between partners is subject to personal or organizational privacy requirements. Private data should be kept confidential.

3.2.5. Solution

The solution depends on sharing a principal's identity information (called federation metadata) between the parties participating in a federation. Federation metadata describes information about federated services, policies describing common communication requirements, and brokering of trust and tokens via security token exchange (issuances, validation, etc.).

Further, establishing security relationships is fundamental for the interoperation of distributed systems. Applying relevant policies is needed to make it clear to the users what is allowed or which conditions apply to the use of web services. Without applying relevant policies and trust relationships between the involved parties, web services have no means to assure security and interoperability in their integration and may lose their ability to provide service.

The value of establishing a federation is to facilitate the use of a principal's identities across trust boundaries to establish a context for that principal.

Structure

Figure 5 describes the structure of this pattern. The Pink component (**Principal**) describes logical web service principals representing service providers and requesters; the yellow (**Security Token Unit**) describes security tokens and claims needed to access resources; the light blue (**Security Token Service Unit**) describes the token management structure; the orange (**Reference Monitor**) represents policy classes to verify tokens and claims; the maroon (**Domain**) represents domain and resources; the grey (**Identity Federation**) depicts the federation of several partners that have established business relationships; while the green (**Identity providers** and **Authorization Services**) are specialized forms of Security Token Services (STS) that provide identity management and make authorization decisions. The naming relations between components describe nature of interaction.

The implementation section shows details of these components according to the WS-Federation standard.

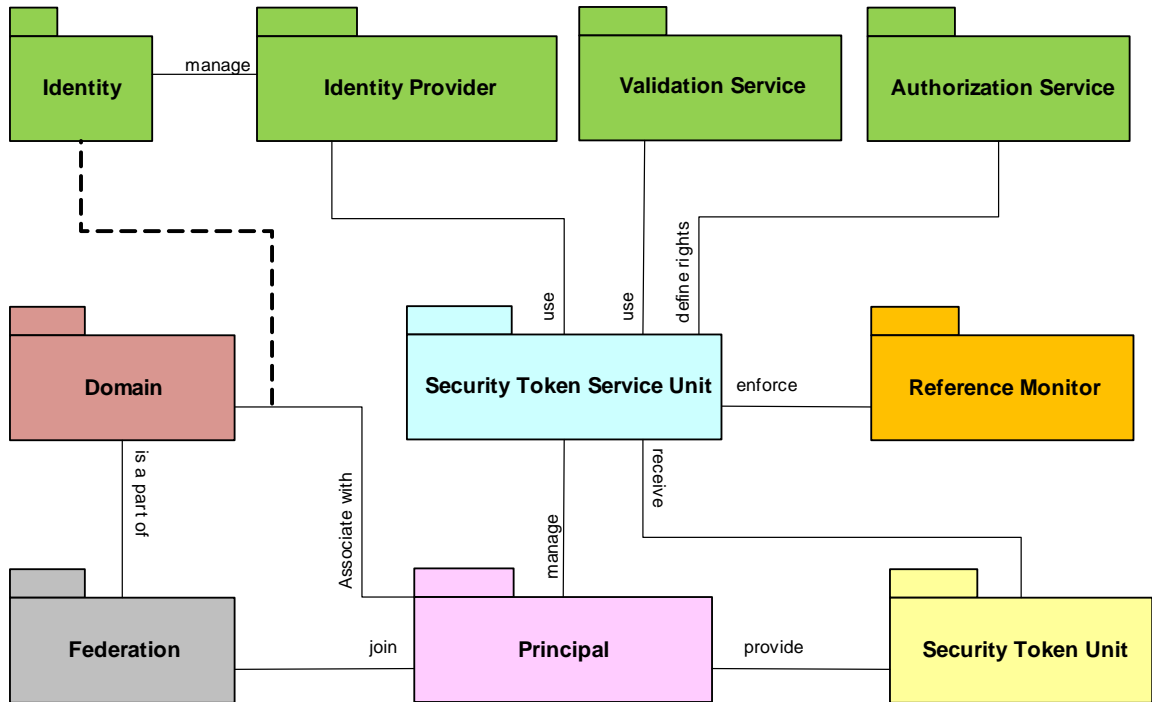


Figure 5: Component diagram for the WS-Federation Pattern

3.2.6. Implementation

Figure 6 expands the units of Figure 5 and describes the class diagram of the WS-Federation standard. We matched the colors of both figures for consistency. The Pink component describes a logical web service connection; yellow describe security tokens and claims, light blue describe the token management structure, orange denotes reference monitor of trust and policy to verify tokens and claims, maroon represents domain and resources the partners willing to access, grey depicts the federation of several partners, while green are specialized forms of Security Token Services (STS).

Principal represents an entity (e.g., an end user, an application, a machine), which can have the role of: **Service Provider**, the one who owns and provides a web service, or **Requester**, who uses federation services to conduct activities using web services.

A **Claim** is a statement about a client, service or other resource (e.g. name, identity, key, group, privilege, capability, etc.). Claims are assertions such as “I am Adnan”, “my ID number: XYZ3014, I am an Account manager at TouchDownVacations.com and I need to reserve a room at Ajiad.com”. Federation partners must agree upon types of claims allowed in the tokens they exchange. A **Security Token** is a collection of claims responsible for adding signatures to tokens. Security Token also is a generalization of: **Signed Security Token**, which is cryptographically endorsed by a specific authority (e.g. an X.509 certificate or a Kerberos ticket). The **Proof-of-Possession (PoP) Token** contains a secret *data* parameter that can be used to prove authorized use of an associated security token. Usually, the proof-of-possession information is encrypted with a key known only to the recipient of the POP token.

Attribute Service maintains information about principals within a federation. Pseudonyms are aliases used at different federations. Pseudonym Service is a service that handles alternative identity information about principals within a federation. In other words, a pseudonym service allows a principal to have different aliases at different domains.

The **Security Token Service (STS)** issues SCTs by itself, or relies on another trusted STS to do so using its own trust statement. It produces assertions based on evidence that it trusts, provides challenge for requesters to ensure message freshness (the message has not been replayed and is currently active), verifies authorized use of a security token, and establishes trust among a domain of services. Each **STS** has a **Trust Engine** that evaluates the security-related aspects of a message using security mechanisms and includes a policy to verify the requester's assertions. The Trust Engine verifies security tokens and verifying claims against policies. A **Policy** is a collection of policy assertions that have their own name, references, and ID.

A **Domain** is a collection of **Resources** and represents a unit of security administration. A **Federation** is a collection of domains that have established business relationships in which one domain can grant authorized access to its resources based on an identity, and possibly associated attributes, that are asserted in another domain (A Circle of Trust). Federation metadata describes settings and information about how a service is used within a federation. Federation metadata location is expressed through Metadata Endpoint Reference (MEPR). Given the metadata endpoint reference for the target service allows the requestor to obtain all requirement metadata about the service (e.g. federation metadata, communication policies, WSDL, etc.).

A **Federation Context** is the group of domains where principals present security tokens, obtain session credentials, and establish associations with domains, forms a **Federated Context**. This federated context is dynamic, in that if the principal has not present security tokens for any domain, that specific domain is not part of the federated

context. Compared to federation, a federated context is a technological decision and is not persistent (In other words, does not exist after the principals decide to terminate it), while federation is a business decision, where principals might form a federation (a group, or a consortium), but it doesn't necessarily mean they have to involve in business interactions.

The **Identity Provider** is a trusted entity used by the requester and the service provider. It issues and manages an **Identity** which is a set of credentials for each subject that will be verified by the controller of accessed resources. The Identity provider is in charge of implementing "Identity Mapping", that is the conversion of a digital identity from one domain to another digital identity valid in another domain. **Authorization Service** is a specialized form of a STS that takes authorization decisions. **Validation Service** is a specialized form of a STS that validate provided tokens and evaluate their level of trust using the WS-Trust mechanisms. Identity Provider, Authorization Service, and Validation Service could be parts of one STS, or parts of different STSs.

Dynamics

Sequence diagrams for two use cases “*Obtain access to a resource using an identity token*” and “*Obtain an access to a resource using a pseudonym ID*” are used to describe the dynamic aspects of the WS-Federation.

Obtain access to a resource using identity token and rights (Figure 7):

Summary: The requester obtains an identity security token to access a resource.

Actors: Requester, identity provider, and validationService (which is a STS).

Precondition: The validationService has a TrustEngine and a policy to verify the requester claims and the requester provides parameters in form of *claims* and *RequestType* signed by a *signature*.

Description:

- a. A requestor obtains an identity security token from its identity provider.
- b. The requester presents this token to the validationSTS for the desired resource.
- c. If successful, the validation service returns an access token to the requestor.
- d. The requestor then uses the access token to access the resource.

Post condition: The requester has an identity access token that can be used to access a resource.

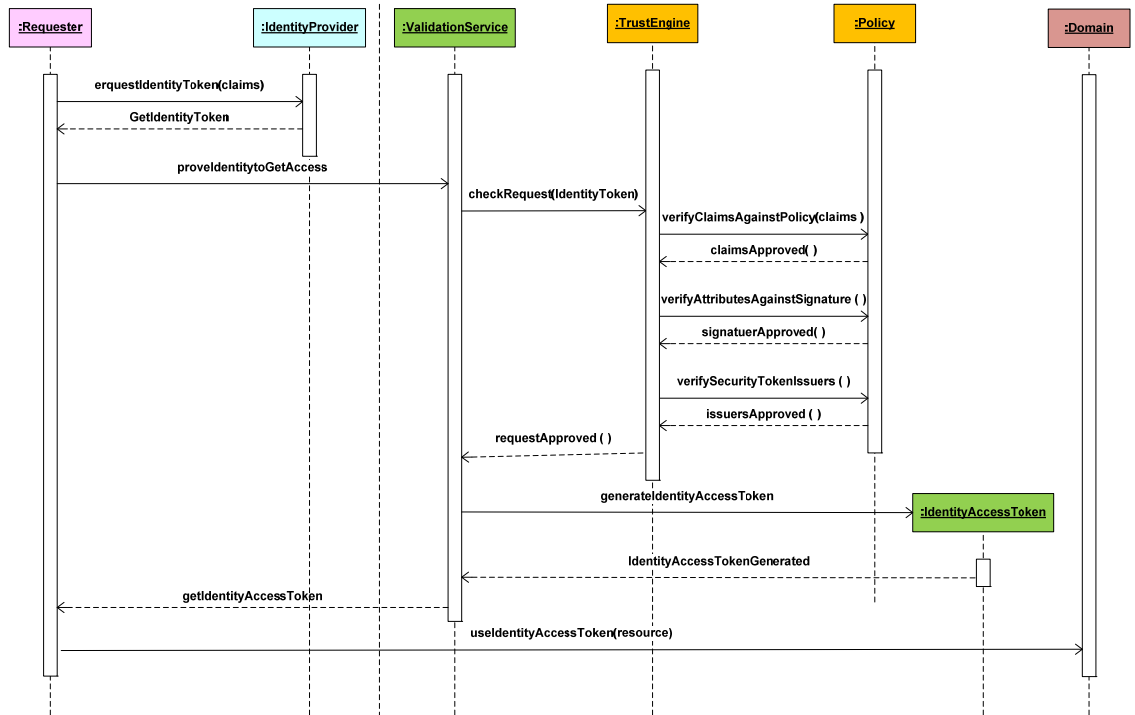


Figure 7: Sequence Diagram Obtain access to a resource using identity token

Accessing a resource using a pseudonym ID (Figure 8):

Summary: The requester accesses a resource using her local identity token; the resource has its own pseudonym service that matches requester's identity with its local identities.

Actors: Requester, Resource, IdentityProvider, PseudonymService (Which is a STS).

Precondition: The resource has its own identity provider and pseudonym services.

Description:

- The requester obtains a token from the resource identity provider.
- The requester asks to access service using identity token.

- c. The resource uses its own pseudonym service to get the local identity for the requester.
- d. Once retrieved, the resource grants access and sends it back to the requester.

Post condition: The requester is granted access to use a resource.

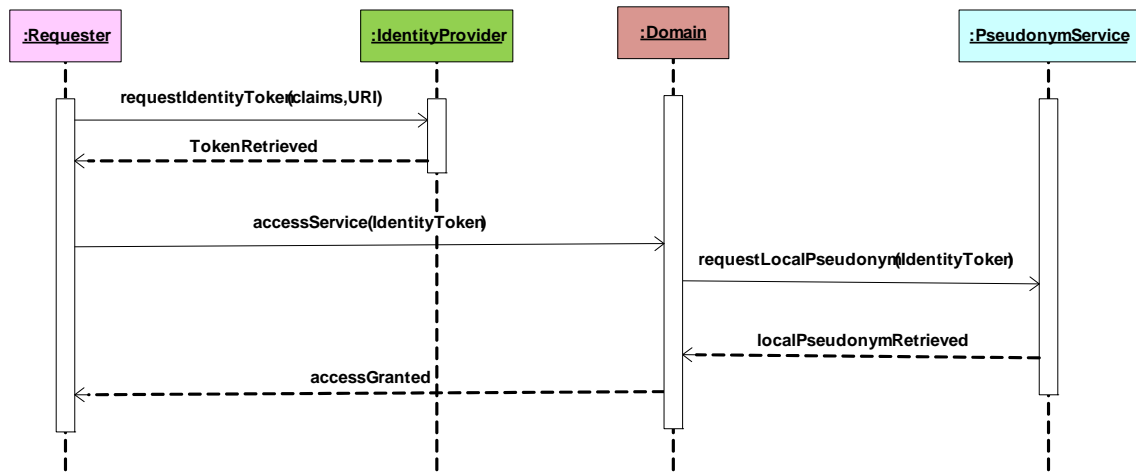


Figure 8: Sequence Diagram accessing a resource using a pseudonym ID

3.2.7. Implementation

In order to assure effective implementation, we need to take in consideration the following:

- We can present a principal's digital identity in different forms requiring different types of mappings. A digital identity is fixed (consistent across domains within a federation). Another approach to identity mapping is pair-wise mapping, where a unique digital identity is used for each principal at each target service. A third approach is to let the requester

generate a digital identity, and let the target service use the requester's mapping service (which is a pseudonym service) to map the given identity into a constant digital identity.

- Different environments will have different configurations based on their needs, security policies, technologies, and existing infrastructure. IdentityProvider, AuthorizationService, and ValidationService are general forms of Security Token Service (STS). We can combine those forms (One STS performs the entire job of all) or separate them (a different STS for each form).
- WS-Federation defines a federation sign-out mechanism. The purpose of federated sign-out is to indicate to federation participants that a particular federation is being terminated and they may want to clean up any cached state or security tokens for a principal that are no longer required because the principal's session is being terminated. Federated sign-out is different than token cancellation as defined in WS-Trust since federated sign-out applies to all tokens and all target sites for the principal within the federation.

It's important to note that no complete security solution for web services is guaranteed through WS-Federation by itself. WS-Federation should be used in conjunction with other web services standards such as WS-Security, WS-Trust, and WS-Policy for an optimal solution.

3.2.8. Example Resolved

By utilizing WS-Federation mechanisms, our travel agency is able to provide access to web services without the overhead of managing other partners' users. This reduces administrative costs (because the accounts for the employees of their partners are handled by the partners themselves) and provides improved service for partners. As each partner employee account is managed by the user's employer, the accuracy of user's attributes asserted in claims is greatly improved because partners know the most current status of their employees. This, in turn, improves security when it comes to access control decisions based on the most up-to-date user context with no worry about orphaned user accounts associated with former users of other partners.

3.2.9. Known uses

The following products have implemented WS-Federation:

- *Active Directory Federation Services (ADFS)* is a standards-based service that allows the secure sharing of identity information between trusted business partners [Mic12].
- *EmpowerID* is an Identity Management and Cloud Security product built on a Business Process Automation (BPA) platform, whose major functions includes user provisioning and Cloud single sign-on [Emp12].
- *IBM Tivoli Federated Identity Manager* provides web and federated single sign-on (SSO) to end users across multiple applications [Tiv12].

- *RadiantOne Cloud Federation Service*, enables a secure federated infrastructure, and creates one access and audit point to connect all internal identity and authentication sources to cloud applications [Rad12].

3.2.10. Consequences

The WS-Federation pattern presents the following advantages:

- ***Identity Mapping***: Partners within a federation don't need to register and maintain other users' identities, and the user is spared from having to get and remember a new login in order to interact with the business. This is done through mapping trusted information about a foreign user (e.g., users from business partners) into authentication and authorization information usable by another partner's resources.
- ***Identity Centralization***: To reduce the cost and duplication of effort of identity management; each partner's identity is almost always managed by a trusted partner.
- ***Degree of Security***: Partners can develop offline operating agreements with other service providers to agree about architecture and privacy policies. They can use mechanisms provided by other web services specifications such as WS-Security to secure access to the policy, XML Digital Signature [Xss08] to authenticate sensitive information and WS-Metadata Exchange [Wme09] to describe what other endpoints need to know to interact with them.

- **Interoperability:** By providing required credentials, and agreeing upon privacy policies, partners could have their own federated identity that is gradually and transparently created to be used within a federation. Web services standards—including SOAP, XML, WSDL, and UDDI—successfully enable developers to create web service solutions that are interoperable across multiple platforms, programming languages and applications.
- **Privacy:** While obtaining federated identity within a federation, partners can classify some of their attributes as private, therefore an identity provider can identify which attributes it shouldn't transmit to other parties. Which attributes are considered private and which are not, depends on the user preferences about the use of their own data.

The WS-Federation pattern presents the following liabilities:

- The WS-Federation standard is a lengthy document with a lot of details that were left out to avoid making the pattern too complex. Somebody interested in addressing more details can check the WS-Federation Standard web page.

3.2.11. Related Patterns

- WS-Security [Has09], defines how to secure SOAP messages applying XML security standards such as XML Encryption and XML Signature.
- WS-Policy standard [Aja10b], describes how to express requirements that are needed or supported by a web service. For instance, it can indicate that a specific signature algorithm must be used when signing a document.

- WS-Trust standard of web services [Aja10b] provides a framework for requesting and issuing security tokens, and to broker trust relationships. It uses WS-Security to transfer the required security tokens, using XML Signature and Encryption to provide confidentiality. This standard may use WS-Policy to specify which security tokens are required at the target.
- WS-SecureConversation for web services [Aja12] describes a standard to allow security context establishment and sharing through the lifetime of the communication session between web services.
- A pattern language for Identity Management [Del07] proposed a pattern language of three patterns for identity management systems. The Circle of Trust that represents a federation of service providers that have trust relationships, the Identity Provider, which centralizes the administration of an organization's users, and the Identity Federation allows the propagation of a user's attributes between different security domains.

3.2.12. Conclusion

We have presented a pattern for the WS-Federation that describes how to manage and broker the trust relationships in a heterogeneous federated environment, including support for federated identities, sharing of attributes, and management of pseudonyms. WS-Federation completed the pattern diagram of Figure 1.

In our future work, the pattern diagram will give us the opportunity into investigate and analyze how WS-Federation and other web services standards fit together.

4. RELATING WEB SERVICES SECURITY STANDARDS

Web services standards are rather complex and verbose and it is not easy for designers and users to understand their key points. Our approach is to express web services security mechanisms and standards as patterns. In this way we can verify if an existing product implementing a given security mechanism supports some specific standard [Fer06c]. Inversely, a product vendor can use the standards to guide the development of a product. By expressing standards as patterns, we can compare them and understand them better. For example, we can discover overlapping and inconsistent aspects between them. A standard defines a generic architecture and this is a basic feature of any pattern; it can then be confirmed as a best practice by looking at products that implement the standard (and implicitly the pattern).

Figure 9 shows a pattern diagram describing the relationships between the patterns for some web services standards. The diagram shows dependencies between the patterns; for example, WS-Security uses policies defined by WS-Policy. Our group has written all these patterns [Ssr13].

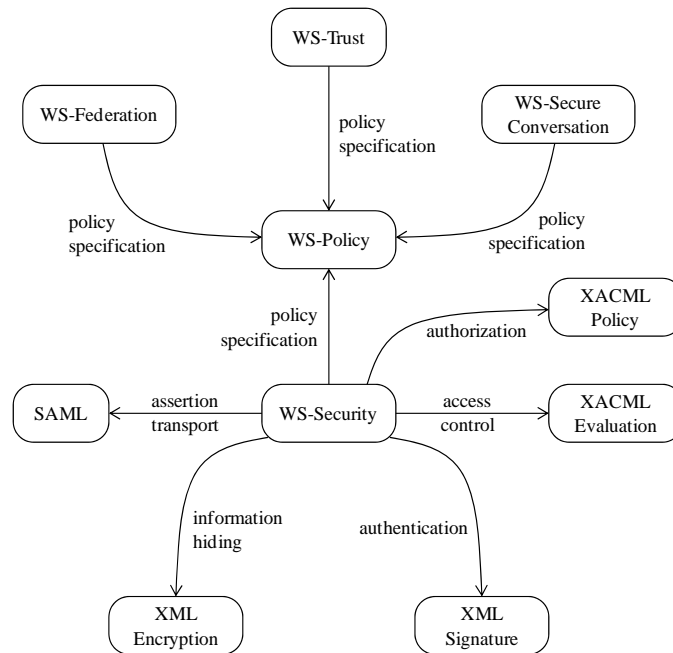


Figure 9: Pattern diagram for web services security standards

4.1. Relationships between Web Services Security Standards

Web services standards offer an approach to security that is:

- *Independent*: from the underlying execution technology and application platforms.
- *Extensible*: to address new requirements and/or exploit new security technologies.
- *Reusable*: Web Services built using web services standards are easy to reuse as appropriate in other services.
- *Flexible*: Can accommodate existing heterogeneous mechanisms that is, different encryption algorithms, different access control mechanisms, and so on.
- *Composable*: Support for composite applications such as business process flows.

- *Interoperable*: Different systems using the same standards can interoperate conveniently.

We show now what types of relationships exist between web services security standards. We will explain the pattern diagram shown in figure 9 in more detail. To summarize all patterns, we are going to list the intent section only of each pattern, and then we will show what degree of dependency a pattern has with other patterns.

Two class diagrams have been presented in previous chapters, for the standards WS-SecureConversation and WS-Federation. We will also show the class diagrams of WS-Policy and WS-Trust, for consistency.

4.1.1. A pattern for WS-Policy

When using web services, it is important to define in advance policies that will apply to interactions with a given web service. These policies can define requirements such as security protocols to be used, expected degree of security or reliability, or other business rules that apply to the specific web service. The WS-Policy standard from W3C provides architecture to define policies and structures, and means to enforce them.

Intent: Without a clear definition of how to use web services, their use could be chaotic. WS-Policy defines a base set of assertions that can be used and extended by other web services specifications to describe a broad range of service requirements and capabilities, including security, reliability, and others. WS-Policy also provides a way to check the requests made by requestors in order to verify that they satisfy their assertions

and their conditions before interacting with the web service. Figure 10 show a class diagram for WS-policy pattern.

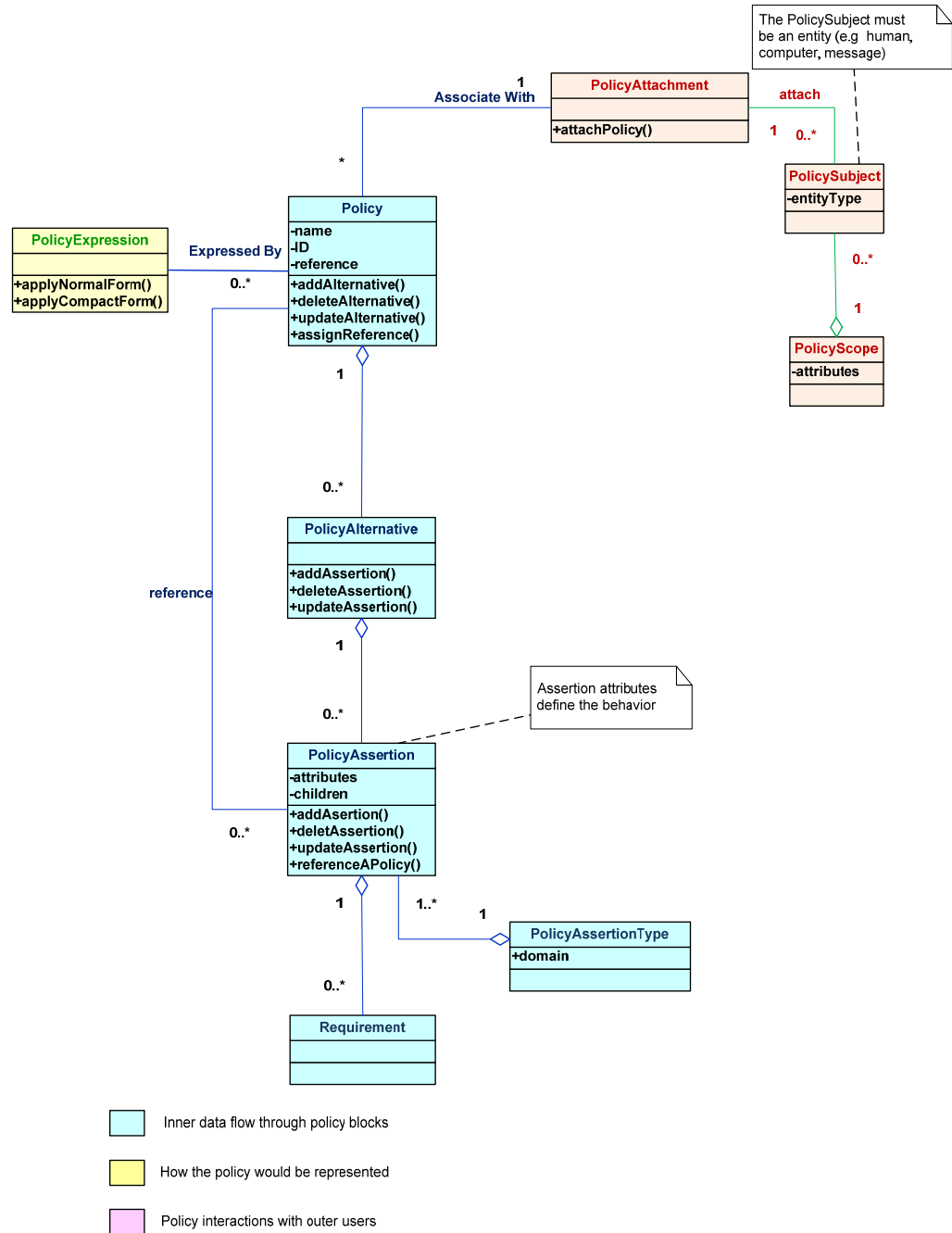


Figure 10: Class diagram for the WS-Policy Pattern

4.1.2. A pattern for WS-Trust

In web services environment, each party needs to determine if they can trust the asserted credentials of the other party. Moreover, the dynamic interaction between the web services requires specifying trust relationships in an explicit way for all parties. Without a clear definition of how web services could manage secure communications and establish trust relationships with other partners, malicious web services could use their business interactions to perform illegal actions. The WS-Trust standard defines how to establish trust between interacting parties.

Intent: WS-Trust defines a security token service and a trust engine which are used by web services to authenticate other web services for which no authentication information exists. Using the functions defined in WS-Trust, applications can establish trust in each other and then engage in secure communication after establishing trust. Figure 11 show a class diagram for WS-Trust pattern

we facilitate the process of communication and increase overall performance. This shared context is implemented by the WS- SecureConversation standard.

Intent: This pattern describes a standard to allow security context establishment and use through the lifetime of the communication session between web services. This security context is used to provide secure communication between web services by extending the mechanisms of WS-Security, WS-Trust, and WS-Policy. Chapter 2 covers the pattern for WS-SecureConversation.

4.1.4. A pattern for WS-Federation

The growth in business-to-business commerce, increased mobility and the importance of persistent interactions between involved parties are some of the current industry challenges. To meet these challenges, companies are extending internal systems to external users of different categories (employees, customers and partners). A variety of users who need to interact with a variety of autonomous systems requires a careful handling of identities. Building secured and trust-based relationships among users might require sharing their identity information. Trust relationships should allow identity and policy data to be exchanged between parties independent of platform, application or infrastructure, and avoid redundant work. A Federation describes the technology and mechanisms necessary to systemize this interconnection, and to allow different domains to use identities from different domains.

Intent: Describe a standard to securely share a principal's identity information across trust boundaries, by having it brokered by identity providers and security token

issuers without the need for re-authenticating users. Chapter 3 covers the pattern for WS-Federation.

4.2. The relationship between WS-policy and WS-Trust

The **Security Token Service (STS)** is a web service that issues security tokens; The **STS** is the heart of WS-Trust and forms the basis of trust brokering. Each **STS** has a **Trust Engine** that evaluates the security-related aspects of a message using security mechanisms and includes policies to verify the requester's assertions. The **Trust Engine** is responsible for verifying security tokens and verifying claims against policies. A main class part of WS-Trust classes is Policy, which can be reused and extended from WS-Policy. A **Policy** is a collection of policy assertions that have their own name, references, and ID. Policies form the basic conditions to establish a trust relationship. Verifying the requester's claims against policy assertions generates an approval to use the target service. A policy may reference another policy (ies), in order to check the tokens sent by the requester or verified by the receiver. Figure 12 identifies the relationship between WS-Policy and WS-Trust.

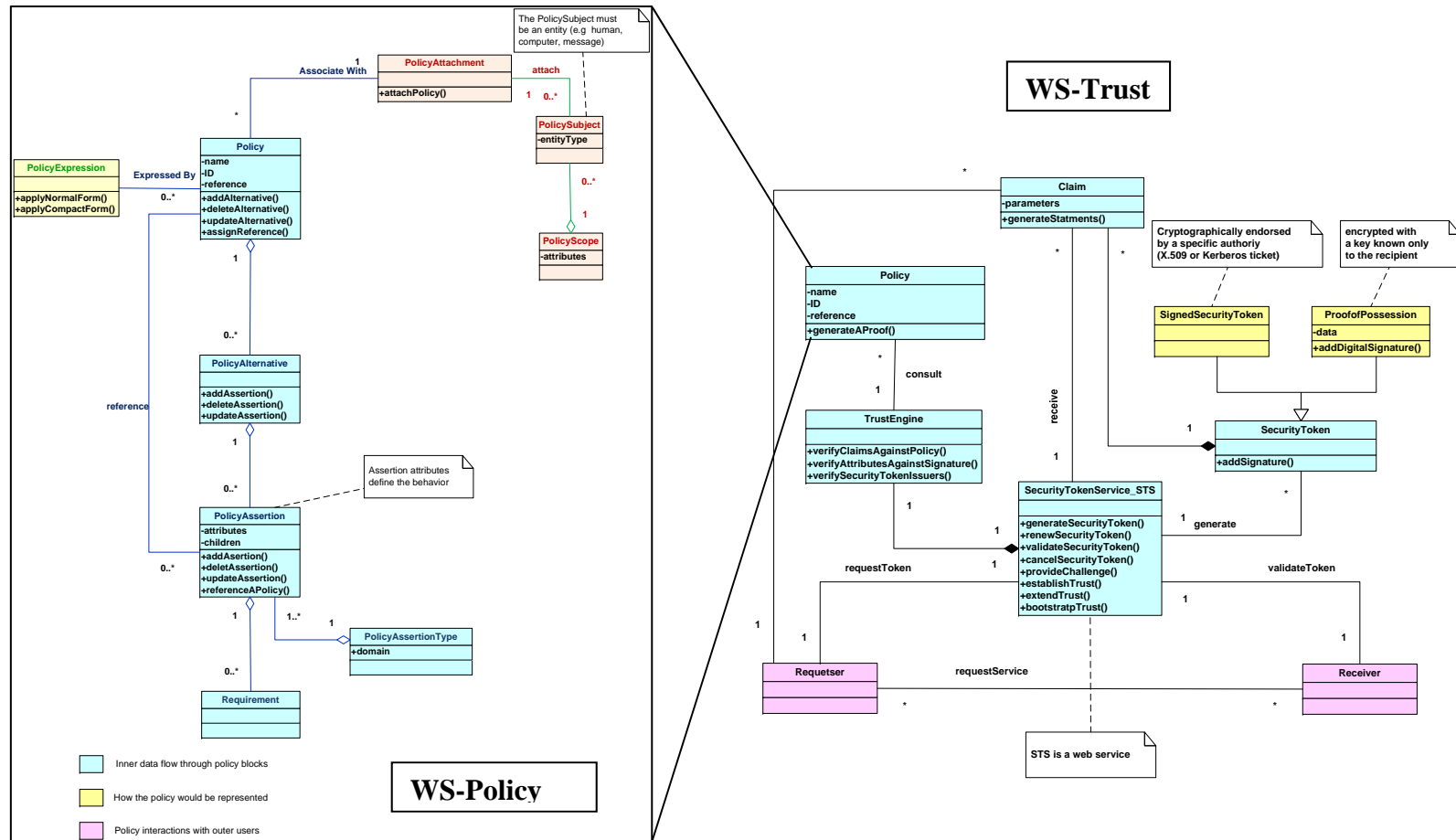


Figure 12: Relationship between WS-Policy and WS-Trust

4.3. The relationships between WS-policy, WS-Trust and WS-SecureConversation

The pattern for WS-SecureConversation defines explicitly an artifact that uses a Security Context Token (SCT). The SCT defines what kinds of assertions are required to be satisfied by any interaction between the involved web services and encapsulates the claims and information sent by the requester in order to obtain the required SCT.

Figure 13 show that WS-SecureConversation uses the services of WS-policy and WS-Trust patterns. The pattern for WS-SecureConversation has a **Security Token Service (STS)** which is a web service that issues SCTs by itself, or relies on another STS to do so using its own trust statement. Two main classes of WS-SecureConversation are: TrustEngine (which can be obtained, reused, and extended from WS-Trust pattern), and Policy (which can be obtained, reused, and extended from WS-Policy). Both TrustEngine and Policy do the same job as explained in section 4.2.

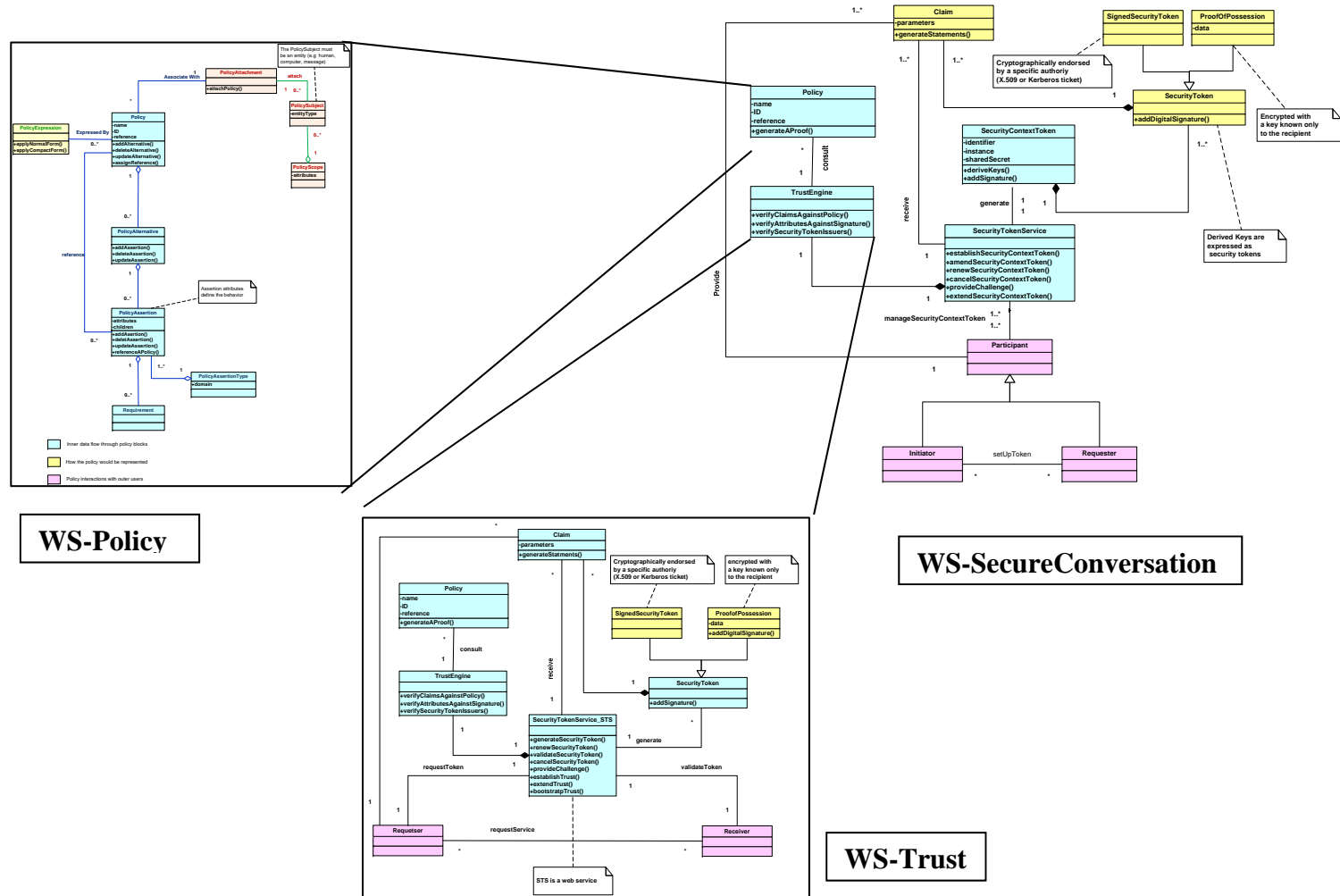


Figure 13: Relationships between WS-Policy, WS-Trust and WS-SecureConversation

4.4. Relationships between WS-policy, WS-Trust, WS-SecureConversation and WS-Federation

The WS-Federation standard is built on top of the WS-Security, WS-Trust, and WS-Policy standards to define a framework with additional federation mechanisms that extend these specifications. WS-Trust is a standard to support the establishment of trust relationships between web services. WS-Policy provides specifications that describe the capabilities and constraints of the security (and other business) policies on intermediaries (for example, required security tokens, supported encryption algorithms, and privacy rules) and how to associate policies with services and end points. Even though WS-Federation is not explicitly extending the WS-SecureConversation, still the WS-Federation can implicitly use the services of the WS-SecureConversation as a part of its STS class.

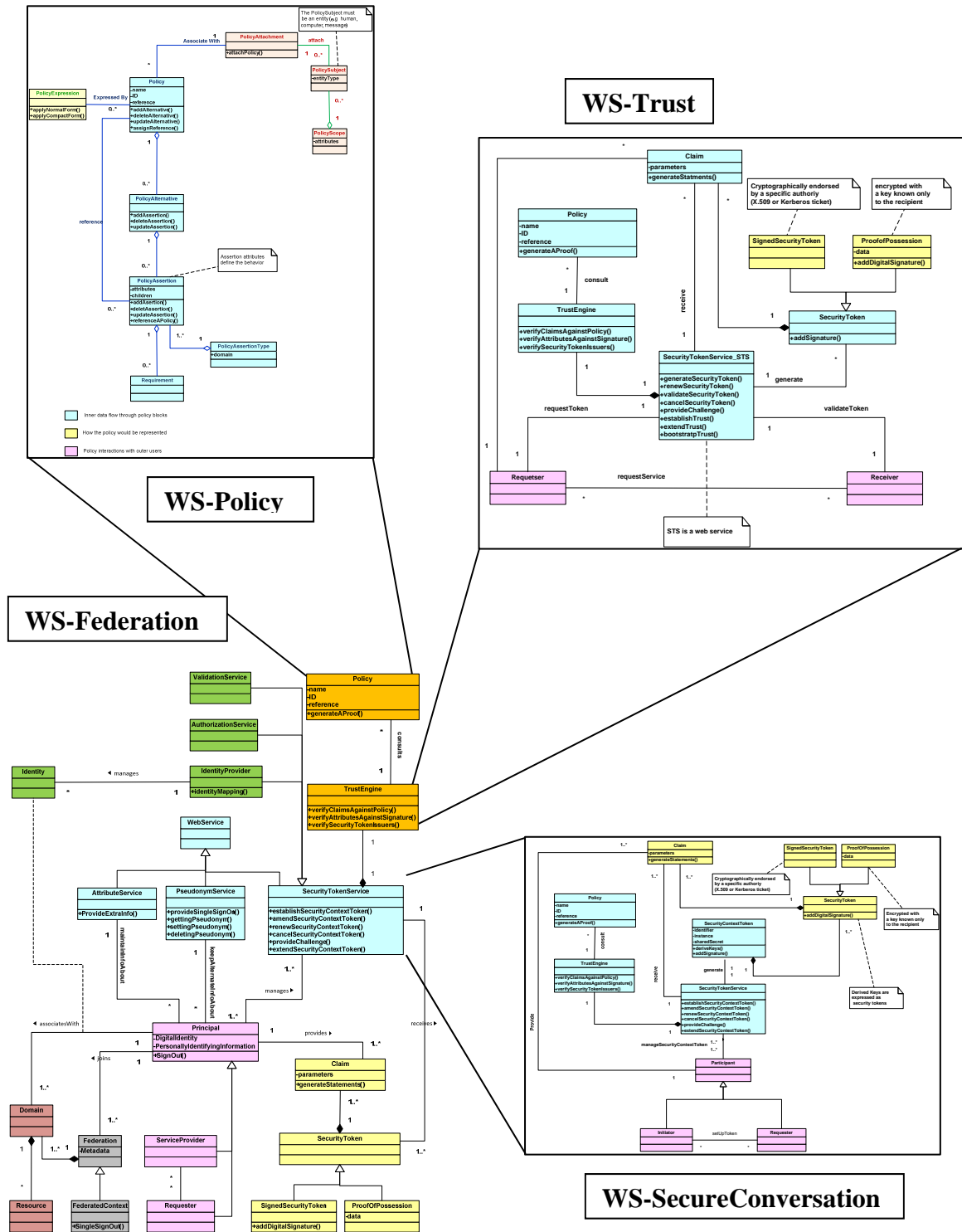


Figure 14: Relationships between WS-Policy, WS-Trust, WS-SecureConversation and WS-Federation

5. ADDING SECURITY TO BPEL WORKFLOWS OF WEB SERVICES

Abstract

BPEL (Business Process Enterprise Language) is a language for web services composition and several implementations of it exist. For BPEL to be effective, it is necessary that it provides more support for security. BPEL doesn't present any means to specify security constraints for workflows. BPEL through its activities tries to provide specific functional aspects and any non-functional aspects are expected to be addressed by other (lower-level) specifications. We present here a way to specify security requirements in BPEL. Since BPEL describes workflows, we present its activities using UML activity diagrams, where we apply a threat enumeration approach to determine the required security mechanisms to stop these threats. Our approach goes beyond BPEL and can be applied to BPMN and other business flow languages.

5.1. Introduction

BPEL (Business Process Enterprise Language) is intended to coordinate interactions among the web services that participate in an application workflow. BPEL as standardized by OASIS [Bpe07], [Wik13] is the most popular language for web service composition. This is supported by the fact that many software companies have integrated

BPEL orchestration engines into their products. Its value for cloud applications is further increasing its use. We need to combine web services from different providers in order to create more advanced and collaborative web services. Two aspects are applied to represent the service workflow, one takes care of which web services participate in the interaction and in what order (control flow), and the other deals with the data been transferred between these interactions (data-flow). There are three main activities involved in an interaction: <invoke> for invoking an operation from one of the partner web services, <reply> to send a response for the requestor, and <receive> to receive a request from the requestor.

For most applications it is critical to consider non-functional aspects in these compositions, such as security, reliability and other aspects of Quality of Service (QoS). With the development of many web services standards such as WS-Security [Wss04], WS-Policy [Wsp07], WS-Trust [Wst07] and WS-Federation [Wsf09], more concerns are addressed to cover non-functional aspects for application specifications in order to encourage the companies to adopt web services in their business activities [Cob11].

BPEL includes some low-level aspects and it is better to specify process structuring in a more abstract way. Two common notations used for process modeling are the BPMN and the UML. BPMN is a modeling notation aimed specifically at business process modeling [Omg13]. UML is a general-purpose modeling notation that was originally developed for designing and specifying software-intensive systems, but which is being increasingly used in other areas, including business process modeling. We adopt

UML due to its ability to allow all the application views to be captured and modeled using a single modeling language thus avoiding the need for different notations to be used within a system for process modeling and technical activities. We will use in particular Activity diagrams and Sequence diagrams to cover a part of the big picture implied in a BPEL process.

A few approaches exist to specify BPMN security but they lack some important aspects. We propose a model based on our previous work on threat enumeration. We developed an approach for security requirements elicitation based on misuse activities [Fer06b], and [Bra08] improved it later to include threat analysis by adding two aspects; the security attribute subverted (confidentiality, integrity, availability and accountability), and the source of threats. By applying these improvements, the approach became more effective, since several more threats can be found. In this paper we are applying this approach in order to be able to add security constraints to workflows. Our contribution stems from finding a better way to add security to workflows, which differently from other approaches, defines security specifications without the need for security specialists. We show our ideas by example, the formal basis for our approach can be found in [Fer06b] and [Bra08].

The remainder of this paper is organized as follows. Section 5.2 explains and illustrates composition in BPEL. In Section 5.3 we introduce an example of a collaborative business process. In Section 5.4 we present our approach of capturing threats using activity diagrams and sequence diagrams. Section 5.5 explains some

techniques to stop or mitigate the identified attacks. In Section 5.6 we discuss related work, summarize the paper and consider future work.

5.2. Background

BPEL specifies a service composition as a process, which declares the web services participating in the composition (partners), data containers (variables), and a set of activities with specific patterns of control and data flow. The building blocks of BPEL processes are activities. There are primitive activities such as `<receive>`, `<invoke>`, and `<reply>` and structured activities such as `<sequence>` and `<flow>`. Structured activities manage the order of execution of their enclosed activities. BPEL processes can run on any BPEL-compliant orchestration engine. The engine orchestrates the invocations of the partner web services according to the process specification.

For illustration, we present a skeleton of the BPEL process that corresponds to a travel agency request to reserve a room in a hotel. For conciseness, we omit some parts of the code.

Listing1.RequestToReserveRoom process

```
<process name="requesttoReserveRoomProcess"/>
```

```
<partnerlinks>
```

```
    <partnerLink name="supplier".../>
```

```
    <partnerLink name="bank".../>
```

```

    <partnerLink name="HotelCompany" .../>

</partnerlinks>

<variables>

    <variable name="clientrqst" messageType="orderInMT"/>

    <variable name="clientrspse" messageType="orderOutMT"/>

    <variable name="payrequest" messageType="payInMT"/>

    ...

</variables>

<sequence name="Main">

    <receive partnerlink="HotelCompany" operation="order" variable="orderqst"
.../>

    <invoke          partnerlink="supplier"          operation="putOrder"
inputvariable="supplyrequest" .../>

    ...

    <invoke partnerlink="bank" operation="pay" inputvariable="payrequest" .../>

    ...

```

```

    <reply partnerlink=" HotelCompany" operation="order" variable="clientrspse"
.../>

</sequence>

</process>

```

5.3. An example for a collaborative business process

An example from the area of travel will be used to illustrate the security issues arising when we define a BPEL process that would be shared across many partners.

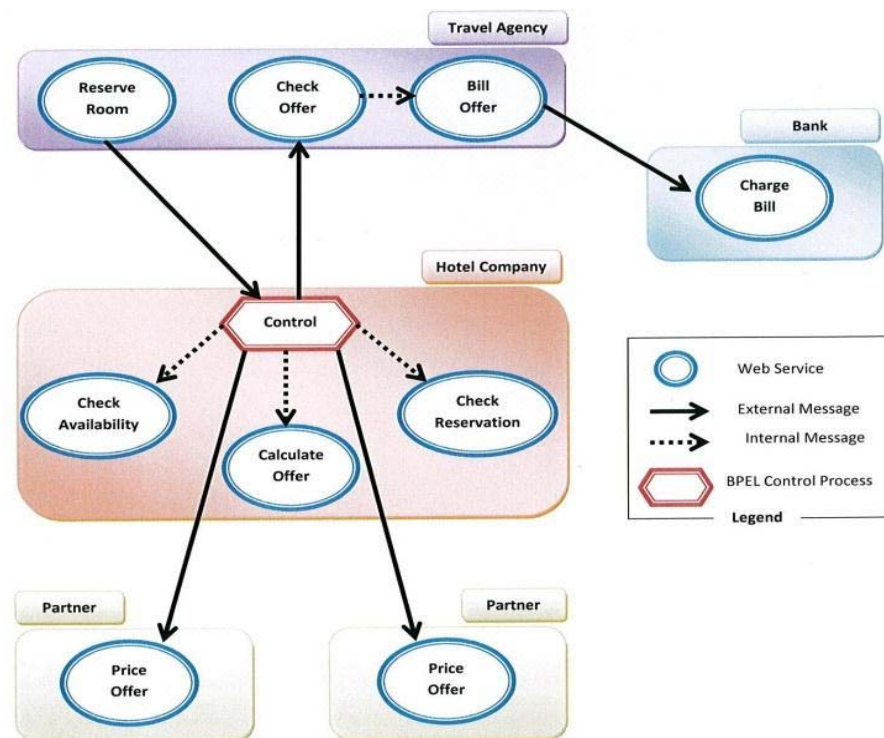


Figure 15: An example of a collaborative business process

The distributed business process shown in Figure 15 is for a travel agency with a chain of offices around the world. The travel agency has a partnership with a hotel

company and a bank and communicates with them using web services. The travel agency defines a BPEL process to serve its customers. This process will reserve rooms from the hotel company and invokes the bank's payment web service to pay for the transaction. More specifically, a BPEL process **Control** is executed in a hotel company. Customers send their request of reservation to the travel agency. A **ReserveRoom** web service of the travel agency invokes the web service offered by the **control** process at the hotel, which provides a list of rooms to be booked. Before placing a reservation, the travel agency expects a price offer accompanied by a commitment with respect to the reservation date.

Figure 16 shows the UML activity diagram for booking a room for a customer. We indicate "swimlanes" for Customer, Travel Agency and Hotel Company. These actions result in new information, including objects for the customer's itinerary, her confirmation, and her invoice. In Figure 17, all the messages between the involved parties indicate the behavior of the web services shown in Figure 16.

When the customer requests a Reservation, the Travel Agency forwards it to the Hotel Company, which invokes a **checkAvailability** service for checking the availability of the rooms in some locations passing it to the list of reservations. After checking availability in several locations, several lists of rooms that match are retrieved from partners. The Hotel Company now invokes the **priceOffer** web services of its sub-partners and provides the respective list of items to each of them.

The **priceOffer** web service checks the availability of rooms on the list in order to return a list of prices and availability on specific locations. The Hotel Company then

invokes a *calculateOffer* web service of the hotel. For this purpose, the control process passes the lists returned from the partners to the *calculateOffer* web service for final prices.

The *calculateOffer* web service identifies the proper request of the travel agency's reservation by processing the data passed by the *checkAvailability* web service. Finally, the offer is returned to the *control* process and will be passed to *checkOffer* web service of the travel agency, which in turn returns an 'OK' or 'Reject' response to the *control* process. Upon accepting the offer from the hotel company, the *billOffer* web service invokes the *chargeBill* web service for the bank to charge the customers the required fees and notify the travel agency to finalize the process.

The response from the *checkOffer* web service is passed to a *createReservation* web service by the Hotel Company. Depending on the type of response, this web service either completes the reservation processing within the hotel if the response was 'OK' or discards the whole request. By doing so, the web service has accomplished its task. It returns a corresponding result to the *control* process that in turn provides this result to the *reserve* web service of the travel agency as a response to its own invocation, thereby completing the workflow of this business process.

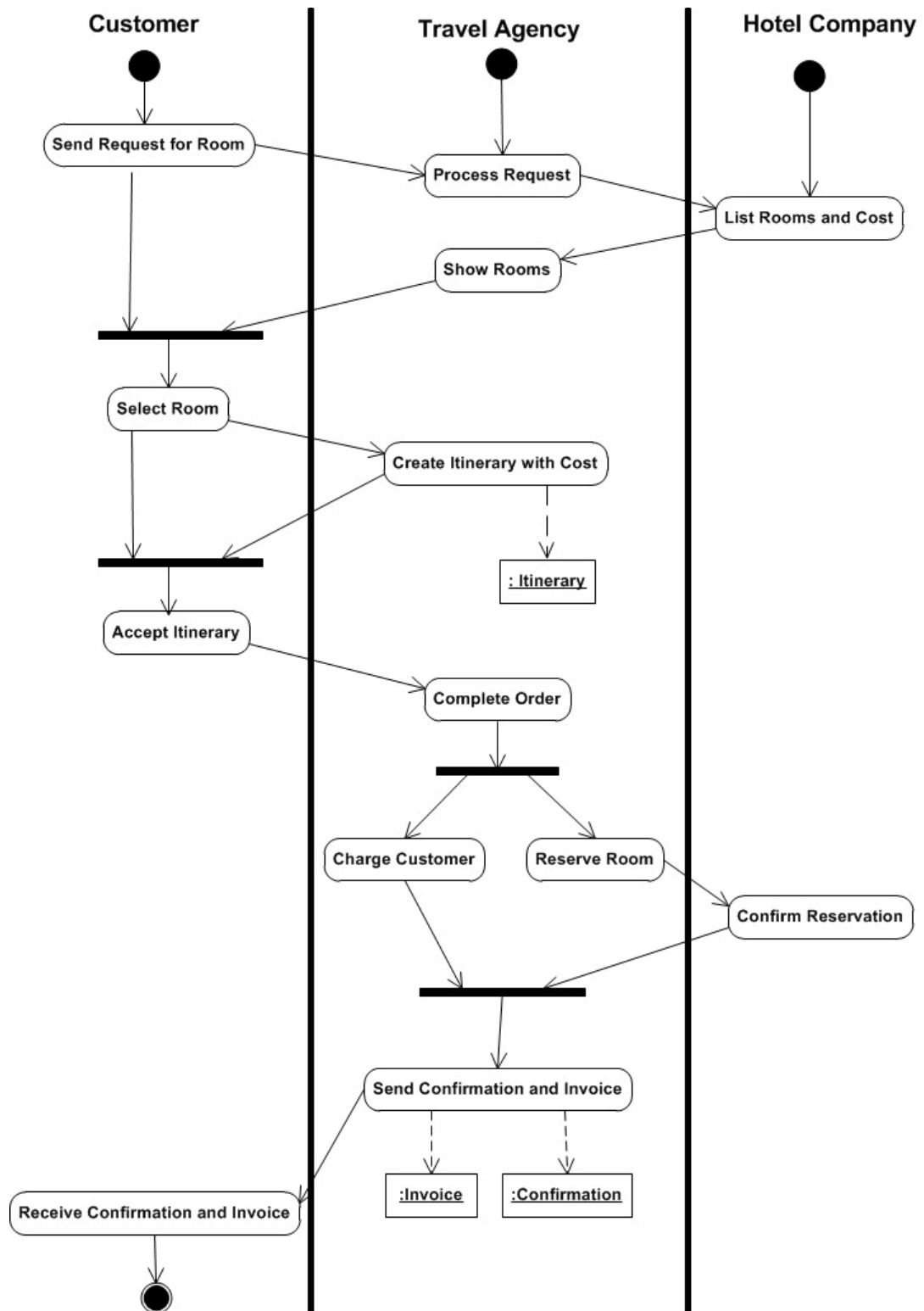


Figure 16: A BPEL Activity diagram for reserving a room

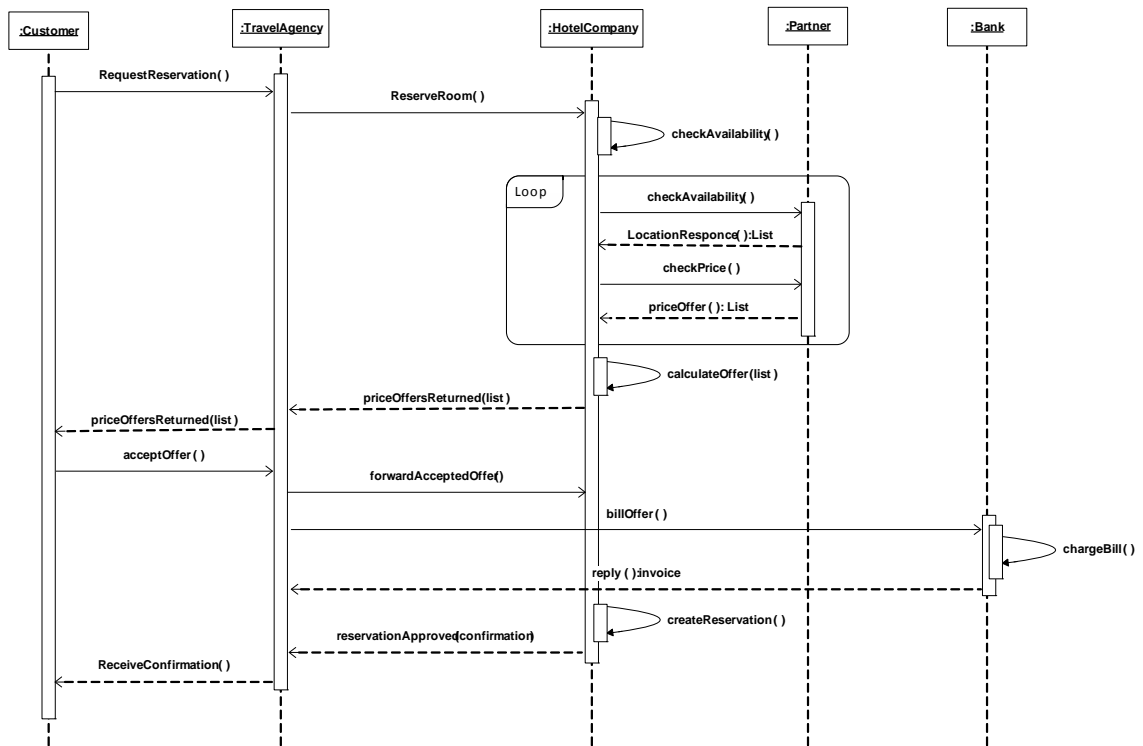


Figure 17: A BPEL sequence diagram for reserving a room

5.4. Threats to the activities

Each activity is potentially susceptible to attack, although not necessarily through the computer system. Figure 18 shows the same activity diagram with some possible threats. The attacks are presented as threats (dotted lines). Undesired consequences in the form of additional or alternative objects (dotted lines) have also been added.

We only show some of the threats; in particular, for the last activities:

- Threat T1 (Illegal dissemination). The travel agency collects the customer's itinerary and uses it illegally.

- Threat T2 (Charge Spurious Fees). The travel agency charges the customer spurious fees.
- Threat T3 (Sends Spurious Ticket and Invoice). The travel agency sends spurious ticket and/or invoice.

Note that:

- We can list systematically all (or most) possible application threats. While completeness cannot be assured, we can consider at least all important possible attacks. The threats that we postulate come from our experience, from the knowledge of the application, and from the study of similar business processes (many online shopping processes have similar threats).
- We can identify internal and external attackers. The actors in these attacks could be external attackers (hackers). It is also possible that a person in a legitimate role can be malicious (internal attacks). For example, Threat T1 and Threat T3 are performed by insiders, while Threat T2 is performed by either external or internal attackers.
- We are not restricted to analyze each activity in isolation. Some workflows require several activities, e.g. "Cancel Reservation" could be followed by "Refund to Customer". We can consider attacks that take advantages of this sequence, for example, by bypassing some steps that perform checks. These threats, in general, are harder to find.

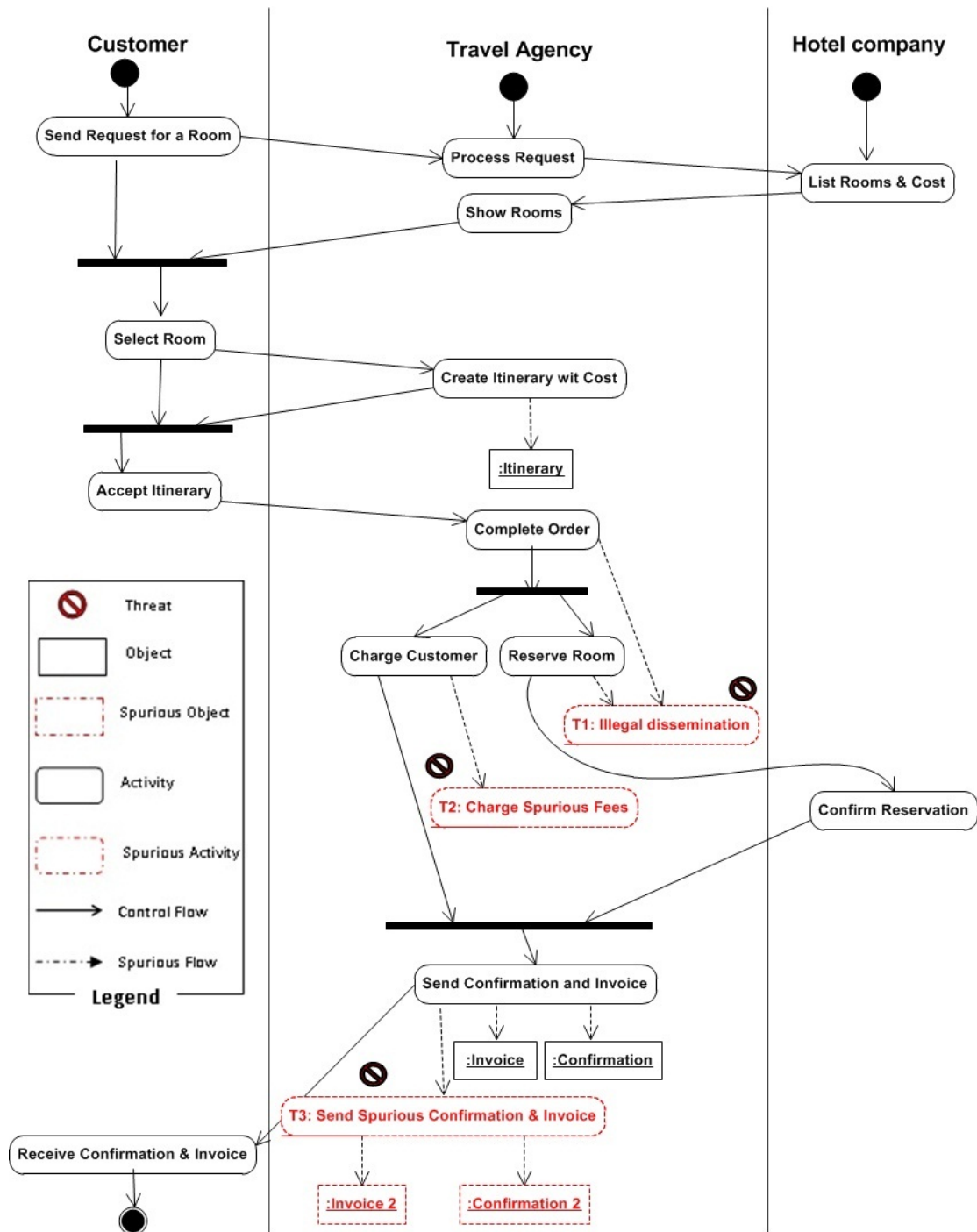


Figure 18: Some threats to the BPEL Activity diagram for reserving a room

5.5. Stopping or mitigating the threats

After we enumerate systematically the threats, we can now find out what policies are needed to stop these attacks. For this purpose, we can select from the typical policies used in secure systems [Fer13], [Gol08]. This selection should result in a minimum set of mechanisms instead of mechanisms piled up because they might be useful. For example, to avoid impostors we can have a policy of I&A (Identification and Authentication) for every actor participating in a business process. Table 1 shows the specific threats and policies for the last activities of Figure 18.

Table 1: Mitigating or stopping threats to a business process using security policies

Threats		Description	Security Policies
Threat 1	Illegal dissemination	The travel agency collects customer's itinerary to use it illegally.	Logging
Threat 2	Charge Spurious Fees	The travel agency charges the customer spurious fees.	Protection against denial of service.
Threat 3	Sends Spurious Ticket and Invoice	The travel agency sends spurious ticket and invoice.	Separation of administration from use of data

The security policies are chosen to mitigate or stop the threats defined in the activity diagrams. By analyzing more activities in the activity diagram, we are able to capture more threats, for which other security policies are needed. These policies are realized using security mechanisms, using appropriate metamodels as shown in [Fer11].

In that approach, security mechanisms are realized as security patterns, where a security pattern is an encapsulated solution to a recurrent security problem. A policy may be realized by one or more patterns.

We can complement this analysis by considering any operation from the partners (Hotel Company, Bank) should require ***authentication***, since it's not acceptable that anyone who knows the web service link could have the ability to reserve a room or perform bank transactions. For that reason, the web service composer has to conform to the partner's ***security policy*** before writing down the BPEL process that will invoke those services. To participate with the BPEL process, the security policy for hotel and bank web services must define which security model and mechanism (certificate, encryption algorithm, digital signature, etc.) they support.

Now that the partner web service can verify the identity of the requestor, the next step is to decide what the requestor is allowed to do. For ***non repudiation*** issues, it's important that the office which did the reservation cannot deny doing so and that nobody can misuse its identity for malicious activities. To fulfill this requirement, ***digital signatures*** can be used. Another issue is ***data integrity***; we need to make sure that when the offices perform the reservation process, it's mandatory that nobody can modify the reservation and change its data. Appropriate mechanisms are also needed to avoid replay attacks, where attackers try to copy the reservation order and resend it again. For that kind of problems, a ***timestamp mechanism*** is applicable.

For the bank's payment web service, we give more importance to *confidentiality* since sensitive data transferred between the travel agency and the bank should not be read by unauthorized people. The agreement to choose an encryption and decryption mechanism is a good choice.

Figure 19 summarizes this discussion and addresses the security considerations for the travel agency as a whole. Since some of the parties involved in this BPEL are external parties, the indicated mechanisms should be considered as expected requirements. On the travel agency case, the selection of those considerations depends on the degree of external and internal communication, and on the sensitivity of the data being transmitted. Note that this approach indicates only the security mechanisms needed in each unit; we need to continue refining the model to indicate which specific web services require which type of security mechanisms following a general approach we proposed in [Fer06a].

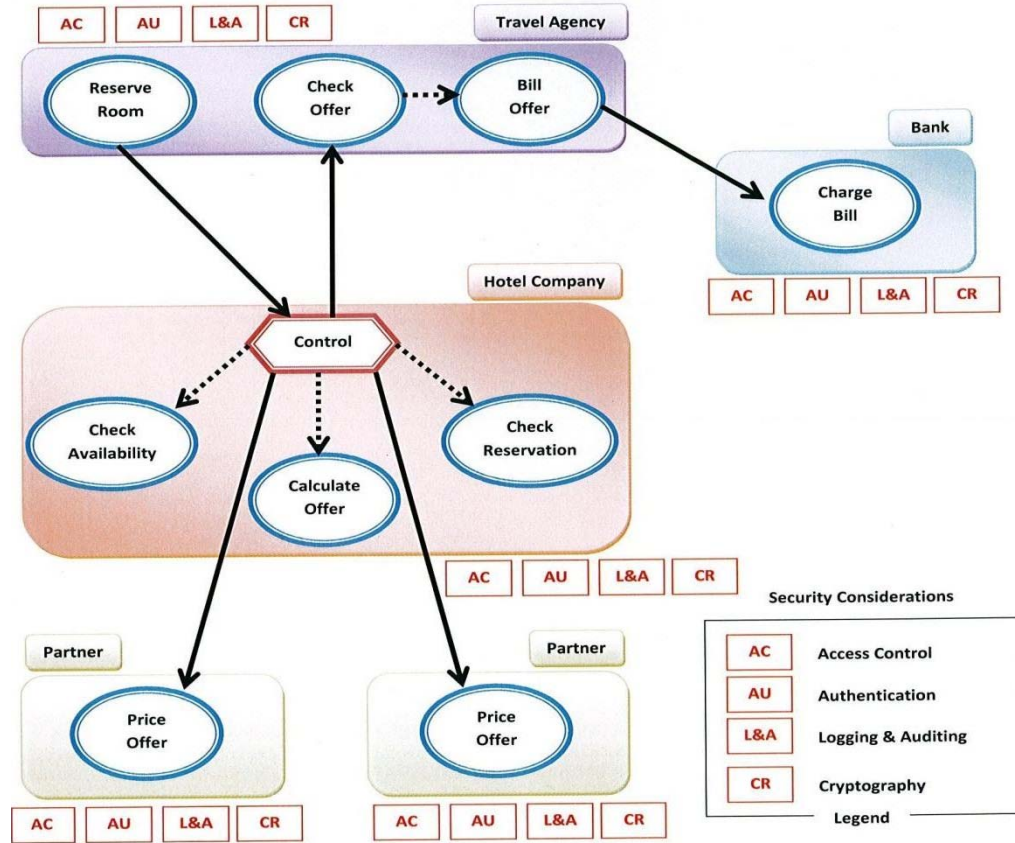


Figure 19: Addressing security considerations for BPEL

5.6. Conclusions and Future Work

We have presented an approach that enumerates the threats to a given BPEL process. We considered UML activity diagrams for collaborative business processes and showed how to list the possible threats and attacks that could happen in order to define the appropriate and suitable countermeasures to stop or mitigate them. The use of UML activity diagrams produces a clear and more intuitive way to analyze these attacks than working directly in BPEL. The first one to suggest making security annotations to activity diagrams was S. Johnston [Joh11] and this idea has been applied in all the

surveyed approaches, including ours. Our claim for improvement lies in that we can enumerate threats without the need of a security expert.

Future work includes a design model for our example of travel agency, which will explain how to deploy web services standards such as WS-Security, WS-Policy and WS-Trust in a systematic way. Such a design model will use our threat enumeration approach to specify exactly which security mechanisms we should deploy for which classes. We intend to incorporate also this approach as part of our secure application design methodology [Fer06a], which now starts from use cases instead of workflows. This would allow building systems combining web services and standard components, which are necessary in real architectures. Our approach starts from use cases and class/sequence diagrams' conversion to BPEL models can be done using an approach such as the one in [Shi05].

6. THE NEED FOR CLOUD STANDARDS

Despite the large amount of active work in developing standards for the Cloud, there is currently a long debate about the role of standards in the Cloud. On one side parties who see the Cloud as a completely new approach that needs a completely new set of standards, and on the other side parties who see the Cloud as a technology built based on existing technologies that already have standards. In this chapter we will define what is a standard and why do we need standards in Cloud Computing?

6.1. The Concept of a Standard

The concept of “standard” can have different interpretations, it might be a public specification issued by a group of companies, to a “de jure” standard issued by a recognized standardization body. The potential users (customers, developers, designers, etc.) can use those different views to get useful indications about how mature and stable a standard is, and what level of endorsement that standard has achieved. There are 3 types of standards, which can be described as in the sections below.

6.1.1. De jure Standards

“De jure” is an expression that means "concerning law" It refers to the standards defined by entities that have a legal status in international or national law. Those entities might be international like The International Organization for Standardization (ISO),

national standards organizations like the British Standards (BSI) in the UK, and the American National Standards Institute (ANSI) in the US, or continental standards such as European standards.

The issuance of a standard by one of these standardization groups is generally a long-lasting process, which might take up years, and the appropriate committee of that standardization bodies should agree on the standard before publishing it. These standards are appropriate for the health and safety areas, in measuring business quality, and for long term IT areas.

6.1.2. De facto Standards

A term used for any product, system, custom, or convention that has achieved a dominant spot by public acceptance or market forces. ." De facto means "existing in fact," or "in practice but not necessarily ordained by law" or "in practice or actuality, but not officially established." [Tfd13].

The term "de facto standard" is used in contrast with mandatory standards (also known as "de jure standards"); or to define the dominant voluntary standard, when there is more than one standard available in the market for the same purpose.

A de facto standard may be endorsed by a standardization initiative, and eventually become a consortium recommendation, or a de jure standard. Relevant requirements are that they are widely used, meet the needs for functionality, and support interoperability.

6.1.3. Consortium Standards

A technology recommended by a group of companies in order to define some functionality. Those groups may vary in size from a few large companies (e.g. IBM, Microsoft, and BEA) to much larger organizations such as the World Wide Web Consortium (W3C), the Organization for the Advancement of Structured Information Standards (OASIS), and the Internet Engineering Task Force (IETF).

A subsequent endorsement by a standardization body will ultimately promote de facto standards to de jure standards, if that de facto standard achieved a higher guarantee of support for interoperability. On the contrary, de jure standards or consortium standards do not guarantee per se that a standard will be broadly endorsed, or the market availability of really interoperable implementations by multiple vendors.

Furthermore, the life cycle of a standard within a standardization body indicates the maturity level of the standard itself., and the definition of a standard and its issuance by a standardization body or by a consortium is considered a long lasting process, subject to that organization procedures; for example, W3C takes 6 months to form a working group on a technology, and then 18 months to 3 years to agree on a recommendation, which is only released if that technology functionalities have proper interoperable implementations, and enough of the members of W3C support it.

6.2. Aspects of a Good Standard

Before going into details in any Cloud standard, let's see what makes a standard valuable. The top three benefits are cost saving, easier maintenance and enhanced security. Consider the case of two Cloud provisioning standards, SPML and SCIM. SPML (Service Provisioning Markup Language) and SCIM (System for Cross-domain Identity Management) solve basically one of the challengeable Cloud issues: federating provisioning of services. For example, let's take a our Ajiad travel agency case, "When I add users to a FavoriteCustomers group in Ajiad Active Directory, I'm going to provision accounts for those users and give them full access to all services designed for that group, and when I remove users from that group, their accounts need to be deactivated with that FavoriteCustomers group only."

The Organization for the Advancement of Structured Information Standards (OASIS) developed SPML and has had two official releases, 1.0 in 2003 and 2.0 in 2006 [Pst13]. SCIM was developed as a collaborative project among vendors including Google, WebEx and Salesforce.com, but now it is under the Internet Engineering Task Force (IETF) [Iet11].

By comparing SPML with SCIM, we found that SPML is more like a real standard that has documentation with clear sections and explanations, whereas SCIM looks a bit more ragtag with no much information of how it came to be or who is using it. Surprisingly, SCIM seems to be more widely used, while there are no major Cloud service providers using SPML [Kup11].

What does really matter when it comes to consider a standard is worth adopting or requiring is, simply, whether the standard is actually in use. Simply formalizing a standard (even if the group in charge is a known standards-developing organization) is not enough to make that standard widely in use. For example, for web-service implementation, REST-based implementation has become in many cases a preferred architecture over SOAP-based implementations (which are more secure) because it is easier to use.

Cloud Computing is presently going through what web services went through back in 2006. It will take some time for standards to emerge. This fact underlines the importance of software architecture for Cloud implementations in which standards-reliant components should be implemented as separate components from the rest of the system in order to reduce the impact of standards evolution.

6.3. The Need for Cloud Computing Standards

These days, most enterprises are still only using the Cloud services to support a small share of their IT needs. Current Analysis published a survey of North American enterprises; *Enterprise Adoption of Cloud Applications and Services* in June 2011 found businesses are using Cloud services to cover fewer than 10 percent of their IT requirements [Cua13].

Two main forces are driving the efforts of emerging Cloud standards: standards-developing organizations who want to use more Cloud services without the fear of any

side-effects of adopting Cloud standards such vendor lock-in, and Cloud vendors who have the mission to remove any blocks preventing customers from using their services.

Vendors are interested in binding users to their own product in order to gain competitive advantages. For that, the goal of providers is mainly focusing on showing they can meet or even exceed customers' requirements, while the goals of enterprises is to ensure that investments in Cloud now will not impact their business in the future.

6.3.1. Facilitate Communications

Using standards to facilitate the communications between two different systems is the most recommended approach for business services integration. There are several convincing reasons for adopting standards. First, standards avoid Cloud users from locking in with any specific Cloud provider; therefore, users will have more power while choosing Cloud services. Second, applications or tools been developed based on standards are more interoperable and can be easily integrated. This allows users (designers, end users, managers, etc.) to take the best approach that matches their needs to build their system architecture. Third, there are more Cloud resources available when widely adopted standards are used. This reduces the cost of recruiting and training IT personnel.

6.3.2. Security

Usually, Cloud customers have their own IT security requirements on mind and they will be asking, what is the best way to evaluate Cloud service providers to make sure they are satisfying our needs?

While there is no absolute answer to that question, there is at least one good place to start from: the Cloud Security Alliance (CSA). The Cloud Security Alliance (CSA) is a not-for-profit organization with a mission to promote the use of best practices for providing security assurance within Cloud Computing [Csa13a]. CSA member organizations in cooperation with other cloud services providers work together to describe best practices in security. The CSA came up with a number of useful resources, many of which have become de facto standards for cloud security.

For example, The CSA Cloud Controls Matrix (CCM) provides a controls framework that gives detailed understanding of security concepts and principles that are aligned to the Cloud Security Alliance guidance in 13 domains [Csa13b]. One strength point to mention that makes CSA CCM so convenient is that it has strong interconnections with industry accepted security standards, regulations, and controls frameworks that may be required in cloud deployments, such as ISO 27001, COBIT, PCI, HIPAA, GAPP and FedRAMP.

CSA offers another useful resource to use which is The CSA's Security, Trust and Assurance Registry (STAR). "STAR" is a free, publicly accessible registry that documents the security controls provided by various Cloud Computing offerings

[Csa13c], and we can use it as a starting point for comparing security features for Cloud vendors. The Cloud service provides access (SATR) to answer questions raised by the CCM. The STAR participants list contains: major IaaS vendors like Amazon Web Services, Hewlett-Packard, Microsoft Windows Azure, SoftLayer and Verizon Terremark, as well as major SaaS vendors like Box.com and Microsoft Office 365.

For the realm of security of Cloud vendors, the work done by CSA is sufficient, but if we jump to any other non-functional requirements, no similar organizations has done a similar job that allow users to compare Cloud vendors in terms of performance or features. The best choice is a comparison services that are far from being standards. This includes InformationWeek Buyer's Guides on IaaS and vendors Cloud Storage, Backup, and Synchronization, and some vendors like Hitachi Consulting and IBM.

The standardization of Cloud Computing is vital and essential to provide a benchmarking foundation. Through the compliance of standards products become more transparent and can be combined to offer more features. This will increase the competition between providers, since they have the opportunity to offer standardized software products independent from the underlying infrastructure at the users' side. A Cloud product built using standards raises higher quality standards due to the assessment of service levels, and offer a higher transparency of services, which lead to an increasing trust of the customer in the provider [Bli03].

6.3.3. Ranks

For comparison purposes, several industries companies attempt to utilize the concept of Cloud Computing and pursuit to enhance Cloud infrastructures. Since there is no universal definition of Cloud Computing, and numerous perceptions of Cloud Computing each of which has its own pros and cons, a lot of companies struggle to make use of the Cloud. To address this demand, several independent providers are providing Cloud service comparisons through published rankings [Tec13], [Cio13].

Unfortunately, these ranks are comparing “apples” with “oranges”, based on varied values with no connections between the dots. For example, for the software and platform levels, these comparisons are apparent and do not cover neither measurable factors (performance, availability, prices, etc.,) nor QoS factors (guarantee, reliability or security, etc.,) in an adequate manner. If we move to the infrastructure level, better ranks exist where benchmarks have been made available [Clh13].

6.3.4. Users Awareness

Users feel a sense of security and confidence when they exactly understand how a process is running and functioning. Even though Cloud Computing looks more convenient for users since they don’t need to worry about understanding processing details, yet, have no chance but to trust the Cloud provider, which raise more concerns and doubts.

Standards are crucial when discussing Cloud Computing security in that a large number of standards, addressing various security aspects have been developed and to date no comprehensive critical analysis of such standards exist. The goal of this chapter is to provide a view of security standards in Cloud and how to connect the dots between those standards.

6.3.5. Comparing Providers

Users nowadays have many choices of Cloud providers to choose from depending on their needs, and the first question users ask when they are looking for a Cloud provider is how do I compare services? And here we can see the urgent to have Cloud standards on hands. For instance, when it comes to CPUs, Cloud vendors have their own metrics for measurement: Microsoft Azure uses the clock speed of its processors, Amazon Web Services has the Elastic Compute Unit, and Google has the Google Compute Engine Unit, but no details beyond that. If the case was comparing security requirements, then the situation will get even more problematic.

6.4. Summary

Depending on what we read and who we believe, Cloud Computing is a revolutionary new paradigm for IT service delivery and belongs at the top of IT vendors' list of concerns. Not surprisingly, confusion also exists about the need for standards in Cloud Computing, which is either urgent, nonexistent, or some wherein between. To make any sense of this, we started this chapter by trying to define what a standard is, and then we explained what makes a good standard. We concluded that what does really

matter when it comes to consider a standard is worth adopting or requiring is, simply, whether the standard is actually in use. We ended the chapter by listing the main factors of why do we need standards for Cloud Computing in a way that's both meaningful and relevant to the current industry.

7. SURVEY OF SECURITY STANDARDS FOR CLOUD COMPUTING

Even though standards have not been a requirement for the vast growth of the Cloud, demands for Cloud standards keep increasing, and numerous standards from different organizations are available in the Cloud market. Nevertheless, the Cloud standardization landscape is so ambiguous because there isn't a central body or forum to control the process of standardization, despite the efforts made many people and organizations on that direction. NIST did a good job of listing the standards relevant to Cloud Computing [Nis13a], but their categorization of Cloud Computing standards is like one size fits all, it is not clear, and it is hard for users and researchers to use.

The motivation of this chapter is to clear up the picture of Cloud standards. We survey general Cloud standards, but we focus more on the standards for security and we classify them by groups of interests. We also add standards that although were not developed for Cloud Computing, have an impact on the use of clouds. This should make exploring Cloud standards easier for both users and researchers.

7.1. Importance of Cloud Computing Security Standards

There are several convincing reasons for having standards for Cloud. First, standards avoid Cloud users from locking in with any specific Cloud provider; therefore,

users will have more power while choosing Cloud services. Second, applications or tools been developed based on standards are more interoperable and can be easily integrated.

This allows users (designers, end users, managers, etc.) to take the best approach that matches their needs to build their system architecture. Third, there are more Cloud resources available when widely adopted standards are used. This reduces the cost of recruiting and training IT personnel. The need for Cloud standards was previously explained in details in Ch.6 of this thesis.

For the goal of revealing how the importance and urgency of Cloud Computing security standards are, IEEE, and Cloud Security Alliance (CSA), published findings of a survey of hundreds of IT professionals, many of whom are actively involved in implementing Cloud projects [Iee13]. Among the survey results, ninety-three percent of respondents said the need for Cloud Computing security standards is important; forty-four percent of those participants mentioned they are involved in development of Cloud Computing standards, and eighty-one percent said they are somewhat or very likely to do so in the next 12 months. Data privacy, security and encryption are the most urgent area of need for standards development followed by interoperability.

IT professionals also see value in well-defined standards for Cloud services, according to the 400 respondents to InformationWeek Standardization Survey: 89% of them rate standards for Cloud infrastructure vendors such as Amazon, Microsoft Azure or

Rackspace as extremely (53%) or somewhat (36%) helpful to their organizations; 85% say the same about software-as-a-service. [Iws12].

The main drivers of standards compliance are: The ISO 27001/27002 Information Security Management Standard, Data Breach Notification, PCI/DSS (Payment Card Industry Standard), EU Data Privacy Legislation, SOX (Sarbanes-Oxley Act) and HIPAA (Health Insurance Portability and Accountability Act).

“Too many cooks spoil the broth” and this is the case with Cloud standards these days. There are too many standardization efforts. The Cloud Standards Wiki [Csw13] list 17 different organization and groups working on Cloud standards.

The failure of putting Cloud Computing standards in a complete picture put their rapid growth in jeopardy. The lack of complete standards could make Cloud Computing riskier to use and restrict Cloud implementations. In addition, it could limit interoperability among Cloud platforms and cause inconsistency in terms of security and interoperability. For example, the lack of standardization will prevent a customer trying to switch from one Cloud platform to another from doing so as effortlessly as switching browsers or e-mail accounts. Lack of standardization makes it difficult for buyers to compare and evaluate Cloud offerings.

As the number of vendors in the Cloud market is increasing, the need for interoperability between Cloud platforms is increasing too [Dow11], [Idc11]. Many organizations besides providers of Cloud services develop their own standards. Those redundant efforts result in inconsistent standards. IT professionals agree with the

researchers that standards are necessary for users to compare different offers that match their needs, reaching independency from suppliers and maximizing their own benefits [Mac09]. So, if we don't have an agreement, in form of a standard, users as well as providers are facing a complicated situation by adopting or offering Cloud services [Bor11].

In a typical Cloud service business, companies and users agree on the service level agreements (SLAs), although those are not sufficiently developed or transparent. If any party decides to transfer Cloud-stored data to a different provider, they might find that they can't do so due to several reasons. One that is common is incompatible APIs, which occurs when the API calls that were used to store the data in a Cloud require the data to be in a certain format that is not compatible or interoperable with the data format required by API calls that a different provider uses to store the data in the Cloud.

Another common problem is caused by a failure to compare data storage formats employed by different providers before it selected a provider to host a Cloud service. (One possible solution is to negotiate with the provider to be more flexible on transferring data to a different provider. This can be done by changing code to the provider's Cloud service API calls.)

In reality, Cloud customers deserve more than just interoperable APIs; they need Cloud service standards to ensure full serviceability for all the Cloud delivery models:

- ***Infrastructure as a Service***: Virtual machines that work on one IaaS provider to be compatible with the virtual machines those work for another IaaS provider.

- ***Platform as a Service***: Platforms that work on one IaaS to be compatible with any PaaS that works on another IaaS. Some platforms may not have the same development tools.
- ***Software as a Service***: Applications been developed on one PaaS to work on a compatible PaaS.

Generally speaking, what does really matter when it comes to consider a standard is worth adopting is, simply, whether the standard is actually in use. Simply formalizing a standard (even if the group in charge is a known standards-developing organization) is not enough to make that standard widely used.

The pace of innovation with Cloud services is becoming fast. Vendors are updating their offerings on a monthly basis, whereas standards organizations usually take years to finalize new releases. Thus, standards today are more likely to start with one or a small number of vendors, and we expect to see more de facto standards (including widely accepted best practices) than formal standards in the Cloud era.

7.2. Issues of Cloud Computing Standardization

Even though standards have not been a requirement for the vast growth of the Cloud, demands for Cloud standards keep increasing. The Cloud standardization landscape is so ambiguous because there isn't a central body or forum to control the process of standardization, despite the efforts made many people and organizations on that direction. The following are the main forces that affect the standardization process of Cloud Computing.

7.2.1. Security

Until now, there is no systematic and standardized way to reflect security requirements on Cloud services [Tak10], despite the efforts made by many initiatives that handle this obstacle, although there are many academic works on this direction. In particular, there is no systematic way to translate security requirements and policies across Cloud providers. To achieve true interoperability, we need to translate specific application and service functionalities from one Cloud to another, and there is no way to do that without standardization. Cloud providers just enumerate their security mechanisms without any reference model, which makes it hard to compare their degrees of security.

Today, there are many standards and widely accepted best practices that outline what safeguards and practices Cloud providers and users need to have in hands to ensure appropriate security. But, having so many standards does not complete the job of defining what constitutes effective Cloud security. Cloud providers and enterprises are left alone, relying on a big list of auditing specifications, regulatory needs, security standards, and third-party attestations to provide some guidance on how to protect their Cloud environments. Apparently, this makes Cloud security even more complicated than it needs to be, and this fragmented approach will not guarantee a sufficient security.

While Cloud providers are putting efforts to prove that they can meet security requirements, these emerging standards are beneficial in helping users choose a Cloud provider that will meet internal and regulatory requirements. Keep in mind that those

security standards are good in the initial selection; once you use a standard to build your own Cloud service, or you choose a provider who meets your needs, and get your service up and running, these standards don't do much work to guide best practices. This is where the use of security patterns appears valuable.

7.2.2. Virtualization

Each Cloud platform may have its own type of hypervisor. The mission of the hypervisor is to manage a host server's processing and other resources so that it can run multiple virtual machines (VMs). What does this have to do with Cloud standards? Well, Cloud systems implementing different hypervisors won't be able to interoperate, since those hypervisors don't use the same data formats. This is true for Cloud platforms also, because their VMs don't interrelate in a systematic way due to the differences of implementations with different network and storage architectures, APIs, databases, and other elements.

Without hypervisor standardization, then only way to move a workload from one Cloud platform to another requires creating a new VM on the second platform and then reinstalling the application, which consumes considerable time and effort.

7.2.3. The Lack of Standardization

The lack of standardization of a relatively young approach such as Cloud Computing is not surprising. . This is proven by the fact that the degree of immaturity

with many standards developed by many organizations prevents one organization from dominating the market and mandates other standards.

“Too many cooks spoil the broth” and this is the case with Cloud standards these days. There are too many standardization efforts. The Cloud Standards Wiki [Csw13] list 17 different organization and groups working on Cloud standards.

The case with Cloud is similar to what happened around 2006 with web services. At one point, there were about 250 specifications, standards, and recommendations to support different quality attributes. Nowadays, there are around 100 standards efforts related to web services. Over time, some of web services standards have become de facto or get broadly supported, such as SOAP, WSDL, BPEL, and WS-Security, while others have simply vanished due to lack of support, such as WS-Privacy and WS-Authorization.

All paradigms struggle when emerging, mainly because of a lack of standardization. Cloud Computing is not an exception, as of today, it does lack standardizations. There is no standardized communication between and within Cloud providers and no standardized data export format which makes it difficult to leave a Cloud provider. The lack of standards also makes it difficult to establish security frameworks for such heterogeneous environments and forces people for the moment to rely on common security best practice.

7.2.4. Lock-In Problem

Enterprises are right to be cautious of adopting Cloud Computing, since today's Cloud providers have much more control over data and user experience than their customers. If a Cloud provider decides to raise its rates or alters their service quality, or in the worst scenario goes out of business, customers may be left hanging there. Some of real-life examples are: Gmail, the most used Cloud email provider, had two major outages in 2012; one of those outages affected 5.25 million users [Per12]. Amazon Web Services, the largest provider of infrastructure-as-a-service (IaaS), had three major outages in 2012 [Whi12], where dozens of major websites that rely on Amazon's Web Services have fallen off the face of the public Web as a result of the outage, including the usual suspects, such as pseudo-social network Pinterest along with check-in site Foursquare and online travel service Airbnb. All of those were commercial sites and the loss is resulting in millions of dollars in direct or indirect losses. But can we imagine what is the loss will be if the sites went down where high-sensitive and critical; such as governmental, military, health facilities or emergency and disaster information services? We leave the answer for the readers.

The problem of "locked-in" restricts users from getting the maximum benefit of investing in Cloud. It is very important to the users to achieve a higher degree of the interoperability between offerings and the possibility of getting services from one Cloud or another. The situation will get more complicated if we add the lack of security standards addressing issues such as data privacy and encryption. With possibly sensitive

information are stored off-site and available only over the Internet, security is a serious concern.

The high competition between Cloud providers adds another dimension to the real difficulty of the lock-in problem in the Cloud; vendor finger-pointing, where every vendor claims that everyone else is locking customers in. This is true and seems like typical vendor finger-pointing, but there is an effort to solve this problem; Fusion PPT, a Cloud Computing strategy and technology company, won a contract from the U.S. Department of Defense to classify Cloud standards and what are the best practices to avoid the lock-in problem. Fusion's strategy emphasizes two areas that can reduce the lock-in issue: interoperability and portability [Hpc12].

7.3. Cloud Standardization Efforts

In order to close the gaps in Cloud service standards, a number of organizations put efforts to work on standards, and to push for more standardization principles in between. We can classify those organizations as shown below.

- Standard organizations that offer approved or working Cloud Computing standards such as Organization for the Advancement of Structured Information Standards (OASIS), and National Institute of Science and Technology (NIST).
- Standard industry organizations that have formed working groups to specify standard interfaces to Cloud Computing, such as the Storage Network Institute Association (SNIA) that has its Cloud Storage Technical Work Group, the

Distributed Management Task Force (DMTF) that formed the Cloud Management Work group, and (IEEE) with its groups P2301 and P2302.

- Organizations directing on developing open Cloud service standards such as the OpenStack Foundation, The Open Group, and Open Grid Forum.
- User advocacy organizations that provide best practices on SLA management such as the Cloud Service Customer Council and the TM Forum.

Table 1 presents a list of Cloud standardization efforts. It provides an indication of the variety, number, and overlap of current projects related to standards for Cloud Computing.

Table 2: Cloud Computing Standardization Efforts

No.	Project Name	Focus
1	CloudAudit, also known as Automated Audit, Assertion, Assessment, and Assurance API (A6)	Secure, open, and extensible interface and methodology used by Cloud providers and consumers to automate the audit, assertion, assessment, and assurance of their environments.
2	Cloud Computing Interoperability Forum	A Common framework for Cloud platforms used to exchange information. Supporting the Unified Cloud Interface Project to build an open and systematic Cloud interface to unify various Cloud APIs.
3	Cloud Security Alliance	Suggested practices for Cloud security. Working on the Security Guidance - Version 3 for Critical Areas in the Cloud.
4	Cloud Standards Customer Council	End-user support group sponsored by the Object Management Group (OMG) and creator of the Open Cloud Manifesto, working on standards,

		security, and interoperability issues related to migration to the Cloud.
5	Cloud Storage Initiative	It is sponsored by the Storage Networking Industry Association (SNIA). It creates and promotes the Cloud Data Management Interface (CDMI). Focusing on adopting Cloud storage as a new delivery model (Data-Storage-as-a-Service).
6	DeltaCloud	Creates an API based on representational state transfer (REST) with a small number of operations for managing instances. It also identifies differences among IaaS providers. It has built libraries for seven providers including Amazon EC2, Rackspace and Eucalyptus.
7	Distributed Management Task Force (DMTF)	It deals with interoperability of management for Cloud systems. It creates the Open Virtualization Framework (OVF). Also runs the Open Cloud Standards Incubator.
8	IEEE P2301, Guide for Cloud Portability and Interoperability Profiles	Provides standardized options for application, interoperability portability, and management interfaces, beside options for file formats, and operation conventions.
9	IEEE P2302, Draft Standard for InterCloud Interoperability and Federation	It Works on protocols for exchanging data, functions, and governance between Clouds. It also works on federation between Clouds.
10	OASIS Identity in the Cloud (IDCloud)	Profiles of open standards for identity management, deployment, and provisioning in Cloud. It is in charge of analyzing risk and threat based on use cases and produces strategies for mitigation.
11	Open Cloud Computing Interface	It develops REST-based interfaces used for managing Cloud resources (computing, storage, and bandwidth, etc.).
12	Open Cloud Consortium	It builds frameworks for interoperability between Clouds. It also operates the Open Cloud Testbed.

13	Open Data Center Alliance	An Independent IT consortium that captures customer vision for long-term data centers requirements. It also develops usage models for Cloud providers.
14	OpenStack	Founded by Rackspace and NASA, it is open source software to run private Clouds. It consists of three core projects: OpenStack Compute (Nova), OpenStack Object Storage (Swift), and OpenStack Image Service (Glance).
15	Standards Acceleration to Jumpstart Adoption of Cloud Computing	For the purpose of creating Cloud standards, it provides use cases that can be supported on Cloud systems. Sponsored by NIST.
16	The Open Group Cloud Work Group	Main duties include working with other Cloud standards organizations to show companies how to integrate Cloud into their organizations.
17	TM Forum Cloud Services Initiative	Developing approaches such as common terminologies, transparency among Cloud providers, and security issues to increase Cloud adoption.

The following is a short summary of the main organizations and groups involved in Cloud Computing standardization process in alphabetical order.

7.3.1. Distributed Management Task Force (DMTF)

DMTF came up with Open Virtualization Format (OVF) that provides a way to move virtual machines from one hosted platform to another. OVF enables simplified and error-free deployment of virtual machines across multiple virtualization platforms [Ovf13].

OVF uses a container that stores virtual machines and its metadata, and enables the migration of VMs among Clouds. It also defines certain features of the VM like size, CPU and networking requirements, memory, and storage and what application that can run on it. However, users must manually handle any details needed for interoperability, such as application-component interoperability.

Enabling a VM to run on multiple platforms, and defining how an application should function in the Cloud to accomplish operations such as session handling and load balancing, are two main areas where the DMTF is still working.

7.3.2. Institute of Electrical and Electronics Engineers (IEEE)

IEEE Working Groups P2301 and P2302 are developing standards that will address management, migration, and interoperability among Cloud Computing platforms. The roadmaps for the two working groups haven't yet established.

P2301, Draft Guide for Cloud Portability and Interoperability Profiles (CPIP), is used to assist Cloud Computing vendors and users in building and using standards-based Cloud products and services, which should lead to increase interoperability, portability and commonality.

P3201 provides profiles of existing and evolving Cloud standards from multiple organizations in critical areas such as Cloud-based applications, portability, management, interoperability interfaces, file formats, and operation conventions. The purpose is to

avoid having multiple standards address the same issues while having no standards addressing others [Pwg13].

P2302, Draft Standard for InterCloud Interoperability and Federation (SIIF), defines the topology, protocols, functionality, and governance required for Cloud to Cloud interoperability and data exchange. [Sii13]

7.3.3. National Institute of Science and Technology (NIST)

Government organizations that move to the Cloud to provide more efficient services to customers, expect de facto federal standards of a Cloud Computing definition. This standard definition is provided by the National Institute Standard and Technology (NIST).

On September 2011, the NIST published the Definition of Cloud Computing in which Cloud Computing is defined as: A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (network, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [Nis13].

The NIST definition lists five essential characteristics of Cloud Computing: on-demand self-service, broad network access, resource pooling, rapid elasticity or expansion, and measured service. It also lists three "service models" (software, platform and infrastructure), and four "deployment models" (private, community, public and hybrid) that together categorize ways to deliver Cloud services.

7.3.4. Open Group

Motivated by the fact that not all IaaS implementations are used to support SOA, the Open Group organization launched a mission to standardize IaaS to support services oriented architectures (SOA) and to help organizations that are building IaaS offerings and SOA. The Open Group published three standards [Opg13]. They are:

- Service Oriented Cloud Computing Infrastructure Framework (SOCCI) is an industry standard for enterprises that wish to provide infrastructure as a service in the Cloud and SOA.
- Service Oriented Architecture Reference Architecture (SOA RA) is standard reference architecture for creating SOA solutions that meet different organizational needs.
- Open Group Service Integration Maturity Model (OSIMM) provides a framework for evaluating the SOA maturity level of an organization.

7.3.5. Open Grid Forum

The OGF, a community of users, developers, and vendors for standardization of grid computing, is developing the Open Cloud Computing Interface (OCCI) which consists of a set of specifications, which describe how Cloud Computing infrastructure service providers deliver their compute, data, and network resource offerings through a standardized interface [Occ13].

OCCI have presented several Cloud Computing management tasks, including deployment, automatic scaling, and network monitoring through multiple protocols and APIs. The development of APIs will ease interfacing and communication among diverse IaaS platforms. The OCCI interface has been used to solve a variety of problems in Cloud Computing such as scientific data processing, drug discovery, cancer research, financial risk analysis, visualization and product design.

7.3.6. OpenStack Foundation

OpenStack Foundation is a non-profit corporate entity that promotes, protects and empowers OpenStack software and its community [Osf13]. OpenStack Foundation mission is to get one IaaS to talk to another. Since Cloud service customers expect an open standard to allow one IaaS provider to fully interoperate with another IaaS provider.

OpenStack Foundation manages “OpenStack”, which is an Infrastructure as a Service (IaaS) Cloud Computing project that is free open source software released under the terms of the Apache License. Developers and Cloud Computing technologists from more than 150 companies (joined as members) work together to create open source Cloud Computing platform for public and private Clouds.

7.3.7. Organization for the Advancement of Structured Information Standards (OASIS)

OASIS has formed two technical committees working on centric Cloud Computing problems. The Identity in the Cloud (IDCloud) Technical Committee is in

charge of resolving security issues regarding identity management in Cloud Computing, it defines gaps in the current identity management standards in order to achieve interoperability between those standards. This will to make sure people using Cloud resources are who they say they are [Idc13].

The Symptoms Automation Framework (SAF) Technical Committee is working on ways to ensure Cloud Computing providers understand consumer requirements, when designing and providing services. SAF facilitates knowledge sharing between consumer and provider which allow both parties to work closely together to ensure maximum quality of service, and reduce cost [Saf13].

7.3.8. Storage Networking Industry Association (SNIA)

The Storage Networking Industry Association (SNIA) organization has been involved with storage standards. The SNIA's Cloud Data Management Interface (CDMI) standard provides a standardized way to exchange data between customers and Clouds, manage Cloud-resident data, and transfer data between Clouds. It gives the client the ability to discover the capabilities of the Cloud storage offerings, use it to specify the interface to access Cloud storage, and to manage the data stored [Cdm13].

CDMI has three main areas to cover; "client to Cloud storage" standard specifies the way a user interacts with Cloud storage, "Cloud to data management" standard handles issues such as QoS and encryption, and finally "Cloud to Cloud" standard focuses on the way of moving stored data among Clouds.

7.4. Survey of Security Standards for Cloud Computing

NIST did a good job of listing the standards relevant to Cloud Computing [Nis13a], but their categorization of Cloud Computing standards is like one size fits all, it is not clear, and it is hard for users and researchers to use.

The motivation of this chapter was to clear the picture of Cloud standards. We surveyed general Cloud standards, we focused more on the standards for security, and we classified them by groups of interests. We also added standards that although were not developed for Cloud Computing, have an impact on the use of clouds. This should make exploring Cloud standards easier for both users and researchers.

7.4.1. Problem with NIST Categorization of Cloud Computing Standards

NIST Cloud Computing standards roadmap report [Hog11] published in 2011 documents the fact that wide standards are by now available to support some functions and requirements for Cloud Computing. While most of these standards were developed in support of the pre-Cloud Computing era, such as those implemented for web services and the Internet, they also applicable to support the functions and requirements of Cloud Computing. Other standards were developed specifically for core Cloud Computing functions and requirements, such as virtualization.

The NIST categorization scheme is not meant to be an absolute single hierarchy, since there could be different perspectives to classify the standards. They followed what is called “collaborative tagging” or a folksonomy, which is a system of classification

obtained from the practice and method of collaboratively generating and handling tags to mark and categorize content, according to the suggestions from the NIST Cloud groups.

NIST also created a web page to list all of the standards relevant to Cloud Computing [Nis13a]. NIST gather the highest-level protocols, definitions and standards that are applicable widely to the Cloud Computing use cases identified on its Wiki Collaboration Site [Nis13b]. As the collection grows, NIST's intention is to classify these according to the taxonomical hierarchy defined by the "Reference Architecture and Taxonomy group" and to supplement this categorization using tags to indicate other areas of applicability for a given standard.

The NIST approach of classification didn't reflect the real needs and requirements of Cloud markets. We can achieve that by adding Cloud users to the picture to indicate other areas of applicability for a given standard. The popularity of a given standard among customers indicates a higher percentage of acceptances. For example, some industrial security standards are well-known and acknowledged, but never mentioned in NIST list of standards. What it really matters when it comes to consider if a standard is worth adopting or requiring is, simply, whether the standard is actually used. Simply formalizing a standard (even if the group in charge is a known standards-developing organization) is not enough to make that standard widely used.

Listing all existing Cloud standards in one place may make it hard for users and researchers to explore them. Having one column for categorization is good, but it depends how we categorize those standards, what is listed under interoperability category

for someone might look more like maintainability for someone else. In other words, two people might not agree on one single way of categorization. We prefer classification by groups. Similar work has been done to classify patterns; [Van09] uses a multi-dimensional matrix where each dimension divides the problem space into generally understood concerns. The combination of concerns from multiple dimensions allows patterns to be precisely associated with regions of applicability, and supports developer navigation to find related patterns with relevant information.

Cloud can be implemented in different models using different technologies. Therefore, in our approach, we classify Cloud standards into groups of interests. Each group has a list of standards related to Cloud. Table 3 below lists our findings of security standards relevant to Cloud Computing.

A good point to mention here is that they kept the categories consistent with NIST Cloud Computing definition and with the Reference Architecture and Taxonomy group. These categories will be revised as needed as this taxonomy matures. The last update of the NIST taxonomy was on March, 31, 2011.

Table 1: Survey of Security Standards for Cloud Computing

Cloud Specific Standards						
No.	Name	Last Update	Goal	Developed By	Standard Status	Security Category
1	Cloud Data Management Interface (CDMI) [Sni10]	Apr, 2010	CDMI defines the functional interface that applications will use to create, retrieve, update and delete data elements from the Cloud. As part of this interface the client will be able to discover the capabilities of the cloud storage offering and use this interface to manage containers and the data that is placed in them. In addition, metadata can be set on containers and their contained data elements through this interface.	SNIA	SNIA Technical Position	Storage, Interoperability
2	Cloud Infrastructure Management Interface (CIMI) [Cim12]	Aug, 2012	The Cloud Infrastructure Management Interface (CIMI) standardizes interactions between cloud environments to achieve interoperable cloud infrastructure management between service providers and their consumers and developers, enabling users to manage their cloud infrastructure use.	DMTF	DMTF Standard	Manageability

3	CloudAudit 1.0 [Cla10]	Jul, 2010	Provides an open, extensible and secure interface that allows Cloud Computing providers to expose Audit, Assertion, Assessment, and Assurance (A6) information for (IaaS), (PaaS), and (SaaS) services to authorized clients.	Cloud Audit	Internet Draft	Auditing
4	Common vulnerability scoring system [Cvs12]	Apr, 2012	Provides a method for rating IT vulnerabilities in a manner that helps organizations prioritize and coordinate a joint response to security Cloud Computing vulnerabilities by communicating the properties of the vulnerability.	ITU-T	Final Draft	Monitoring and Incident Response
5	Open Cloud Computing Interface (OCCI) [Occ11]	Nov, 2011	The Open Cloud Computing Interface is a RESTful boundary protocol and API that acts as a service front-end to a provider's internal management framework. OCCI describes APIs that enable cloud providers to expose their services. It allows the deployment, monitoring and management of virtual workloads (like virtual machines), but is applicable to any interaction with a virtual cloud resource through defined http(s) header fields and extensions. OCCI endpoints can function either as	Open Grid Forum	OGF published standards (Proposed Recommendations)	Manageability, Monitoring, Data transfer

			service providers or service consumers, or both.			
6	Open Virtualization Format (OVF) [Ovf13]	Jan, 2013	OVF is a packaging standard designed to address the portability and deployment of virtual appliances. OVF enables simplified and error-free deployment of virtual appliances across multiple virtualization platforms. OVF is a common packaging format for independent software vendors (ISVs) to package and securely distribute virtual appliances, enabling cross-platform portability. By packaging virtual appliances in OVF, ISVs can create a single, pre-packaged appliance that can run on customers' virtualization platforms of choice.	DMTF	DMTF	Interoperability (Virtual machine management)
7	Topology and Orchestration Specification for Cloud Applications Version 1.0 [Tos12]	Nov, 2012	Provides a language to describe service components and their relationships using a service topology, and it provides for describing the management procedures that create or modify services using orchestration processes	OASIS	Public Review Draft	Portability

8	Usage Record (UR) [Ogf07]	Feb, 2007	The Usage Record standard establishes an XML format for exchange of accounting and service usage data in cloud and grid transactions. The format is intended for exchange of data across arbitrary systems at a level of granularity sufficient to merit reporting of computational time, network transactions, or storage. It is oriented toward use in contexts that can aggregate the usage results separately.	Open Grid Forum	OGF Proposed Recommendation	Accounting, monitoring, billing
9	Use Cases for Identity Management in the Cloud [Uci12]	May, 2012	Develops profiles of open standards for identity deployment, provisioning and management in Cloud Computing	OASIS	Working Draft	ID management
10	X.idmcc – Requirement of IdM in Cloud Computing [Rid12]	Oct, 2012	concentrates on how to harmonize the telecommunication services and the Internet services based on a common identity management infrastructure in the Cloud Computing environment	ITU-T	Standard	Identity Management
Web Services Standards						
No	Name	Last Update	Goal	Developed By	Standard Status	Security Category

11	WS-Federation 1.2 [Wsf09]	May, 2009	Defines mechanisms to allow different security realms to federate, such that authorized access to resources managed in one realm can be provided to security principals whose identities are managed in other realms	OASIS	OASIS Standard	ID Mapping and ID Decentralization
12	WS-Policy 1.5 [Wsp07]	Sep, 2007	WS-Policy Defines a framework for allowing web services to express their constraints and requirements. Such constraints and requirements are expressed as policy assertions.	W3C	W3C Recommendation	Security policies
13	WS-SecureConversation [Wss07]	Mar, 2007	Defines extensions to allow security context establishment and sharing, and session key derivation	OASIS	OASIS Standard	Authentication
14	WS-Security Rights Expression Language (REL) Token Profile 1.1 [Rel07]	Feb, 2007	Describes the use of ISO/IEC 21000-5 Rights Expressions with respect to the WS-Security specification	OASIS	OASIS Standard	Integrity and confidentiality

15	WS-Security: Kerberos Binding 1.1 [Ktp07]	Feb, 2006	Describes the use of Kerberos tokens with respect to the SOAP Message Security specification.	OASIS	OASIS Standard	Encryption
16	WS-Security: SAML Token Profile 1.1 [Sam06]	Feb, 2006	Defines a standard set of SOAP extensions that implement SOAP message authentication and encryption.	OASIS	OASIS Standard	Authentication and encryption
17	WS-Security: SOAP Message Security 1.1 [Wss04]	Feb, 2006	Proposes a standard set of SOAP [SOAP11, SOAP12] extensions that can be used when building secure Web services to implement message content integrity and confidentiality, providing support for multiple security token formats, multiple trust domains, multiple signature formats, and multiple encryption technologies. The token formats and semantics for using these are defined in the associated profile documents.	OASIS	OASIS Standard	Message content integrity and confidentiality
18	WS-Security: Username Token Profile 1.1	Feb, 2006	Describes how to use the UsernameToken with the SOAP Message.	OASIS	OASIS Standard	Identification and Authentication

	[Utp06]					
19	WS-Security: X.509 Certificate Token Profile 1.1 [Ctp06]	Feb, 2006	Describes the use of the X.509 authentication framework with the SOAP Message Security specification	OASIS	OASIS Standard	Authentication
20	WS-SecurityPolicy 1.3 [Wsp09]	Feb, 2009	Defines a set of security policy assertions for use with respect to security features provided in Message Security.	OASIS	OASIS Standard	Security policies
21	WS-Trust [Wst09]	Feb, 2009	Uses WS-Security mechanisms and defines additional primitives and extensions for security token exchange to enable the issuance and dissemination of credentials within different trust domains	OASIS	OASIS Standard	Confidentiality and Integrity
22	Simple Object Access Protocol (SOAP) [Soa07]	Apr, 2007	SOAP is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks. SOAP can form the foundation layer of a web services protocol stack, providing a basic messaging framework upon which web services can be built. SOAP is a strongly-typed variant of XML-based communication that	W3C	W3C recommendation	Data Communication

			provides a full description of the required actions taken by a SOAP node on receiving a SOAP message. To resolve ambiguities inherent in the specification, this protocol is generally used according to specific restrictions and clarifications encoded into externally documented profiles. (The use of SOAP in web services settings, for example, is carried out in the context of the WS-Interoperability Basic Profile.)			
23	XML Encryption Syntax and Processing [Xes02]	Dec, 2002	Specifies a process for encrypting/decrypting digital content (including XML documents and portions thereof) and an XML syntax used to represent the (1) encrypted content and (2) information that enables an intended recipient to decrypt it.	W3C	W3C Recommendation	Confidentiality
24	XML signature (XMLDSig) [Xds08]	June, 2008	Signs data—a resource—of any type, typically XML documents, but anything that is accessible via a URL can be signed. An XML signature used to sign a resource outside its containing XML document is called a detached signature; if it is used to sign some part of its containing document, it is called an enveloped signature; if it contains the signed data within itself it is called an enveloping signature.	W3C	W3C Recommendation	Integrity, Non-Repudiation
25	eXtensible Access Control Markup	Jan, 2013	Describes how to interpret the policies. XACML offers a vocabulary for expressing the rules needed to define an organization's security policies and make authorization	OASIS	OASIS Standard	Access Control, Authentication &

	Language (XACML) [Xac05]		decisions. XACML has two basic components: (1) an access-control policy language that lets developers specify the rules about who can do what and when; (2) a request/response language that presents requests for access and describes the answers to those queries.			Authorization, Security Policy Management
26	Security Assertion Markup Language (SAML) SAML2.0 [Sam05]	Mar, 2005	Defines the syntax and semantics for XML-encoded assertions about authentication, attributes, and authorization, and for the protocols that convey this information. The SAML specification covers both the syntax for encoding security assertions (for attribute, authentication and authorization) and the protocols/APIs of how these assertion messages can be exchanged.	OASIS	OASIS Standard	Authentication & Authorization, ID Management
27	Extensible Markup Language (XML) [Xml06]	Sep, 2006	XML is a set of rules for encoding documents in machine-readable form. XML's design goals emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services.	W3C	W3C Recommendation	Data Communication, Data Format
28	XML Path Language (XPath) [Xpl10]	Dec, 2010	XPath is a language for addressing parts of an XML document. It is based on a tree representation and provides methods to navigate, select nodes from, and perform manipulations on the tree elements. While	World Wide Web Consortium (W3C)	W3C Recommendation	Data Communication, Data Format

			there is a 2.0 specification available, the 1.0 subset is interpreted correctly and so can be used by 2.0-compliant implementations.			
29	REpresentational State Transfer (REST) [Res00]	Dec, 2000	REST is an architectural pattern for use of application-layer communications in a manner that uses standards, but is not a standard in and of itself. The primary programming paradigm for the use of REST is that access to a given resource returns a representation of that resource, putting the client application into a state. REST accesses and returned data can take place over any application-layer protocol and are not limited to HTTP.	(None)	Architectural style	Data communication, state transfer
Security Related Standards						
No.	Name	Last Update	Goal	Developed By	Standard Status	Security Category
30	Common vulnerabilities and exposures [Cve11]	Apr, 2011	Identifies high level requirements for enumerating common vulnerabilities that can be used to exchange continuous monitoring cybersecurity information.	ITU-T	Final Draft	Monitoring and Incident Response
31	Computer Security Incident Handling Guide [100]	Jan, 2012	assists organizations in establishing computer security incident response capabilities and handling incidents	NIST	Standard	Monitoring and Incident Response

32	Control Objectives for Information and Related Technology (COBIT) [Coi12]	Oct, 2012	Defines the requirements for the security and control of sensitive data.	ISACA	Final	Auditing
33	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile [Ipk08]	May, 2008	Profiles the format and semantics of certificates and certificate revocation lists (CRLs) for the Internet PKI.	IETF - (RFC5280)	Standard/ RFC5280	Authentication & Authorization
34	International Organization of	Oct, 2005	Provide a model for establishing, implementing, operating, monitoring, reviewing, maintaining and improving the Information Security Management System	ISO	Final	Monitoring

	Standardization (ISO27001) [Iso05]		(ISMS).			
35	Key Management Interoperability Protocol (KMIP) [Kmi10]	June, 2010	Defines a single, comprehensive protocol for communication between encryption systems	OASIS	OASIS	Confidentiality
36	OAuth (Open Authorization Protocol) [Oap10]	Apr, 2010	Allows users to share their private resources (e.g. photos, videos, contact lists) stored on one site with another site without having to hand out their credentials, typically username and password.	OAuth	Standard/RFC 5849	Authentication & Authorization
37	OpenID Authentication [Oid07]	Dec, 2007	Describes how users can be authenticated in a decentralized manner, obviating the need for services to provide their own ad hoc systems and allowing users to consolidate their digital identities.	Open ID community	Final	Authentication & Authorization
38	PCI Data Security Standard [Pci10]	Oct, 2010	Provides an actionable framework for developing a robust payment card data security process -- including prevention, detection and appropriate reaction to security incidents.	PCI Security Standards Council	Standard	Monitoring and Incident Response

39	Statement on Auditing Standards (SAS) No. 70 [Sas92]	Apr, 1992	Defines the standards an auditor must employ in order to assess the contracted internal controls of a service.	(AICPA)	Final	Auditing
40	Secure Sockets Layer (SSL)/ Transport Layer Security (TLS) [Tls08]	Aug, 2008	Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that "provide communications security over the Internet". TLS and SSL encrypt the segments of network connections above the Transport Layer, using symmetric cryptography for privacy and a keyed message authentication code for message reliability.	IETF	Standard/ RFC 5246	Authenti- cation & Authoriz- ation, Confiden- tiality
41	Security Content Automation Protocol (SCAP) [Sca12]	Nov, 2012	Provides guidelines for the development of a continuous monitoring program that provides visibility into organizational assets, awareness of threats and vulnerabilities as well as the effectiveness of security controls.	NIST	NIST Special Publication	Monitori- ng and Incident Respons- e
42	Service Provisioning Markup Language (SPML)	Apr, 2006	Specifies how to exchange user, resource and service provisioning information between cooperating organizations. The goal of SPML is to allow organizations to securely and quickly set up user interfaces for Web services and applications, by letting enterprise	OASIS	OASIS Standard	ID Manage- ment

	[Spm12]		platforms such as Web portals, application servers, and service centers generate provisioning requests within and across organizations.			
43	X.1500 Cybersec urity informati on exchange technique s [Cie11]	Apr, 2011	describes techniques for exchanging cybersecurity information	ITU-T	Standard	Monitori ng and Incident Respons e
44	X.509 Public Key Infrastruc ture (PKI) Proxy Certificat e Profile [Pki04]	Jun, 2004	Defines a standard method of production for proxy certificates, including the ability to support extended attribute certificates conforming to an external profile. Such certificates can be used to convey delegation information and policy restrictions for use of PKI-based credentials in remote settings.	IETF - (RFC3820)	Standard/ RFC 3820	Authenti cation & Authoriz ation
Transport and Network Standards						
No.	Name	Last Update	Goal	Developed By	Standard Status	Security Categor y

45	Domain Name System (DNS)		DNS is a hierarchical naming system built on a distributed database for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most importantly, it translates domain names meaningful to humans into the numerical identifiers associated with networking equipment for the purpose of locating and addressing these devices worldwide.	Internet Engineering Task Force (IETF)	Standard/RFC	Transport, Network
46	File Transfer Protocol (FTP)		FTP is a standard network protocol used to copy a file from one host to another over a TCP/IP-based network, such as the Internet. FTP is built on client-server architecture and utilizes separate control and data connections between the client and server. FTP users may authenticate themselves using a clear-text sign-in protocol but can connect anonymously if the server is configured to allow it.	Internet Engineering Task Force (IETF)	Standard/RFC	Transport, Network
47	GridFTP: Protocol Extensions to FTP for the Grid		Extends the FTP-related IETF RFC 959, RFC 2228 and RFC 2389 standards to include strong authentication protocols, third-party control of data transfer, multiple TCP streams between 2 network endpoints, (including cases in which the number of sending and receiving nodes are different), partial file transfer, manual or automatic control of TCP buffer/window sizes, support for reliable and	Open Grid Forum	OGF Full Recommendation	Data transfer

			restartable data transfer, and integrated instrumentation.			
48	Hypertext Transfer Protocol (HTTP)		The Hypertext Transfer Protocol (HTTP) is a networking protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web.	Internet Engineering Task Force (IETF), World Wide Web Consortium (W3C)	Standard/ RFC	Transport, Network
49	Simple Mail Transfer Protocol (SMTP)		Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (e-mail) transmission across Internet Protocol (IP) networks.	Internet Engineering Task Force (IETF)	Standard/ RFC	Transport, Network
50	The Internet Protocol Suite (TCP/IP)		The Internet Protocol Suite is the set of communications protocols used for the Internet and other similar networks. It is commonly also known as TCP/IP, named from two of the most important protocols in it: the Transmission Control Protocol (TCP) and the Internet Protocol (IP), which were the first two networking protocols defined in this standard.	IETF	Standard/ RFC	Transport, Network
Federal and Governmental Related Standards						
No.	Name	Last Update	Goal	Developed By	Standard Status	Security Category

51	FIPS 140-2: Security Requirements for Cryptographic Modules [Src01]	May, 2001	Specifies the security requirements that will be satisfied by a cryptographic module, providing four increasing, qualitative levels intended to cover a wide range of potential applications and environments.	NIST	FIPS Publication	Confidentiality
52	FIPS 180-4: Secure Hash Standard (SHS) [Shs12]	Mar, 2012	Specifies five hash algorithms that can be used to generate digests of messages.	NIST	FIPS Publication	Integrity
53	FIPS 181: Automated Password Generator (APG) [Apg93]	Oct, 1993	Specifies a standard to be used by Federal organizations that require computer generated pronounceable passwords to authenticate the personal identity of an automated data processing (ADP) system user, and to authorize access to system resources.	NIST	FIPS Publication	Authentication & Authorization
54	FIPS 185: Escrowed Encryption Standard (EES)	Feb, 1994	Specifies an encryption/decryption algorithm and a Law Enforcement Access Field (LEAF) creation method which may be implemented in electronic devices and may be used at the option of government agencies to protect government telecommunications.	NIST	FIPS Publication	Confidentiality

	[Ees94]					
55	FIPS 186-3: Digital Signature Standard (DSS) [Dss09]	Jun, 2009	Specifies a suite of algorithms that can be used to generate a digital signature.	NIST	FIPS Publication	Integrity
56	FIPS 188: Standard Security Label for Information Transfer [Ssl94]	Sep, 1994	Specifies security label syntax for information exchanged over data networks and provides label encodings for use at the Application and Network Layers.	NIST	FIPS Publication	Confidentiality
57	FIPS 190: Guideline for the Use of Advanced Authentication Technology Alternatives	Sep, 1994	Specifies security label syntax for information exchanged over data networks and provides label encodings for use at the Application and Network Layers.	NIST	FIPS Publication	Authentication & Authorization

	es [Gua94]					
58	FIPS 191: Guideline for the Analysis of Local Area Network Security [Gal91]	Nov, 1994	provides appropriate security for local area networks (LANs)	NIST	Standard	Securit y Monito ring and Inciden t Respon se
59	FIPS 196: Entity Authentic ation Using Public Key Cryptogra phy [Eau97]	Feb, 1997	Specifies two challenge-response protocols by which entities in a computer system may authenticate their identities to one another.	NIST	FIPS Publication	Authen tication & Authori zation
60	FIPS 197: Advanced Encryption Standard (AES)	Nov, 2001	Specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information.	NIST	FIPS Publication	Confid entialit y

	[Aes01]					
61	FIPS 198-1: The Keyed-Hash Message Authentication Code (HMAC) [Hma08]	Jul, 2008	Specifies a keyed-hash message authentication code (HMAC), a mechanism for message authentication using cryptographic hash functions.	NIST	FIPS Publication	Integrity
62	FIPS 199: Standards for Security Categorization of Federal Information and Information Systems [Ssc04]	Feb, 2004	Addresses one of the requirements specified in the Federal Information Security Management Act (FISMA) of 2002, which requires all federal agencies to develop, document, and implement agency-wide information security programs for the information and information systems that support the operations and the assets of the agency, including those provided or managed by another agency, contractor, or other source.	NIST	FIPS Publication	Security Policy Management

63	FIPS 200: Minimum Security Requirements for Federal Information and Information Systems [Ms06]	Mar, 2006	Specifies minimum security requirements for federal information and information systems and a risk-based process for selecting the security controls necessary to satisfy the minimum requirements.	NIST	FIPS Publication	Security Policy Management
64	FIPS 201: Personal Identity Verification (PIV) of Federal Employees and Contractors [Piv06]	Mar, 2006	Specifies the architecture and technical requirements for a common identification standard for Federal employees and contractors to achieve appropriate security assurance for multiple applications by efficiently verifying the claimed identity of individuals seeking physical access to Federally controlled government facilities and electronic access to government information systems.	NIST	FIPS Publication	Identity Management and Authentication
65	The U.S. Health Insurance Portability and Accounta	Aug, 1996	Standardize the handling, security and confidentiality of health-care-related data. It mandates standard practices to ensure security, confidentiality and data integrity for patent information.	The U.S. Department of Health & Human Services	Final	Confidentiality, Integrity

	bility Act (HIPAA) [Hip96]					
66	Sarbanes- Oxley (SOX) [Sox02]	Jul, 2002	Defines specific mandates and requirements for financial reporting. It dictates what records are to be stored and for how long.	U.S. Securities and Exchange Commission	Final	Auditin g and reporti ng

7.4.2. Discussion

We primarily focused on identifying the security standards relevant to Cloud Computing. The NIST in [Hog11] did a great a job on that, but as discussed before it is not complete, hard to interpret and read. To close the gap between NIST lists of Cloud Standards and Cloud markets these days, we also added some industrial security standards that are popular in the market these days.

We added a column of “last update” to notate when was that standard updated for the last time. We choose to specify security category on a separate column, since we are more interested into analyzing and modeling security standards for Cloud Computing.

Despite having many organizations working on defining Cloud standards, there are still no accepted standards for Clouds, but NIST is leading some work in this direction [Hog11]. Our analysis of Cloud security standards showed that security standards are verbose, lengthy and use different notations to describe their structures which makes them more difficult to interpret, hard to use and may produce inconsistencies. Some of those Cloud standards are complicated, overlapped or even vague; creating patterns for those standards makes them easier to understand and deploy. This led us to the next section where we described standards using patterns.

7.5. Patterns for security standards of Cloud Computing

Describing standards using patterns makes them easier to understand, and provides guidelines for their implementation in specific systems. For example, Web services are complex and lengthy documents with long descriptions (typically 50-150

pages). We use patterns to model web services standards [Fer13]. Patterns are encapsulated solutions to recurrent system problems in a given context. Patterns define a vocabulary that concisely expresses requirements and solutions as well as providing a communication vocabulary for designers. Out of the 66 standards surveyed here, we found only 9 patterns. Our analysis of Cloud security standards showed that security standards use different notations to describe their structures, which makes them more difficult to interpret and apply. Creating patterns for those standards makes them easier to understand and deploy.

All the patterns described here have been previously published and are part of an ongoing catalog of security patterns [Fer12]. As far as we know, this is the only catalog of web services security standards. However, pattern catalogs are not very useful without a way to apply them to build secure systems. Our group has developed a general secure systems development methodology [Fer06a], which in principle applies also to Cloud Computing.

Table 4: List of patterns for security standards of Cloud Computing

No	Name	Last Update	Security Category	Pattern
1	eXtensible Access Control Markup Language (XACML) [Xac05]	Jan, 2013	Access Control, Authentication & Authorization, Security Policy Management	"Patterns for the eXtensible Access Control Markup Language", [Del05]
2	Secure Sockets Layer (SSL)/ Transport Layer Security (TLS) [Tls08]	Aug, 2008	Authentication & Authorization, Confidentiality	A Security Pattern for the Transport Layer Security (TLS) Protocol", [Kum12]
3	WS-Federation 1.2 [Wsf09]	May, 2009	ID Mapping and ID De-centralization	"A pattern for the WS-Federation of web services ", [Aja13]
4	WS-Policy 1.5 [Wsp07]	Sep, 2007	Security policies	"A pattern for the WS-Policy standard", [Aja10a]
5	WS-SecureConversation [Wss07]	Mar, 2007	Authentication	"A pattern for the WS-SecureConversation standard for web services", [Aja12]
6	WS-Security: SOAP Message Security 1.1 [Wss04]	Feb, 2006	Message content integrity and confidentiality	"The WS-Security pattern", [Has09a]
7	WS-Trust [Wst09]	Feb, 2009	Confidentiality and Integrity	" A pattern for the WS-Trust standard of web services", [Aja10b]
8	XML Encryption Syntax and Processing [Xes02]	Dec, 2002	Confidentiality	Symmetric Encryption and XML Encryption Patterns", [Has09b]
9	XML signature (XMLDSig) [Xds08]	June, 2008	Integrity, Non-Repudiation	Digital Signature with Hashing and XML Signature patterns", [Has09c]

7.6. Summary

The growth of Cloud Computing has been accompanied by an urgent call for standards. Standards play an important role in adopting Cloud Computing, having convenient standards to help customers choose among services, and designers provide a reliable and secure service is a must. The typical reasons have been given: to endorse interoperability, ease the transition of users to Cloud-based services, permit open middleware, and prevent vendor lock-in.

There are so many standards developed by many organizations. In the Cloud, the standardization process is so ambiguous, the reason for that is there isn't a central body or forum to control the process of standardization, despite the efforts made by lots of people and organizations on that direction. The standardization process faces many challenges; security is on top of the list, since until now, there is no systematic and standardized way to constantly reflect security requirements on Cloud services. Virtualization is important too, users are concerned more about transforming smoothly between Cloud providers. Lock-In problem should be taken into consideration.

A number of organizations put efforts to work on Cloud standards, and to push for more standardization principles in between. We have listed the main organizations behind building Cloud standards.

The motivation of this chapter was to clear the picture of Cloud standards. First, we highlighted the problem with NIST categorization of Cloud Computing. Second, then we surveyed general Cloud standards while focusing more on the standards for security, and classified them by groups of interests. Finally, we added standards that although were not developed for Cloud Computing, have an impact on the use of clouds. This should make exploring Cloud standards easier for both users and researchers.

Analysis of Cloud security standards showed that security standards use different notations to describe their structures, which makes them more difficult to interpret and apply. Creating patterns for those standards makes them easier to understand and deploy. We listed patterns for security standards of Cloud Computing which are part of an ongoing catalog of security patterns we are working on.

8. CASE STUDY: SECURITY STANDARDS IN AMAZON WEB SERVICES

Amazon Web Services (abbreviated AWS) is a collection of remote computing services (also called web services) offered over the Internet that together form a Cloud Computing platform to enable customers to build a wide range of applications from enterprise applications and big data projects to social games and mobile apps. The most central and well-known of these services are Amazon EC2 and Amazon S3.

AWS promises customers an end-to-end security. From its side, AWS builds services in accordance with security best practices, offers different security features in those services, and prepare documentations of how to use those features. But it's the user's responsibility to choose the one to use from those best practices and features to build a secure application environment. The ultimate goal for AWS is to ensure the confidentiality, integrity, and availability of customers' data, in order to gain their trust.

In a multi-tenant environment, more concerns are addressed about security, where security should be implemented in every layer of the cloud application architecture. In Amazon Web Services case, AWS is typically handling the physical security, while customers are in charge of application-level security, who should also implement the best practices as applicable to their businesses.

AWS has published its own view of security best practices [Sbp13], through which AWS recommends some basic features and guidelines on how to secure cloud application in the AWS environment. Then users implement additional security best practices using standard methods as appropriate or as seen fit. AWS recommendations are good practices but without any conceptual structure or model, they do not offer any complete or conceptual security.

8.1. AWS SOAP-based web services Vs. REST-based web services

According to AWS documentation, Amazon made it clear that, when it says "web services" it doesn't just mean SOAP-based web services, but REST-based too.

The following table gives an overview of some AWS services and the corresponding available APIs, as stated by AWS taken out of the AWS documentations.

Table 5: An overview of some AWS services and the corresponding SOAP or REST APIs

Service	REST	SOAP
Amazon Simple Storage Service (Amazon S3)	X	X
Amazon Elastic Compute Cloud (Amazon EC2)		X
Amazon Simple Queue Service (SQS)		X
Amazon SimpleDB		X
Amazon Flexible Payments Service (FPS)		X
Amazon CloudFront	X	

8.2. AWS Security Best Practices

To address Cloud customers concerns about security, AWS offer the following security best practices and guidelines on how to secure your cloud application in the AWS environment:

- Protect your data in transit
- Protect your data at rest
- Encryption on operating systems
- Protect AWS credentials
- Manage multiple Users and their permissions
- Secure your Application

In summary, the AWS cloud handles the complexity of the physical security while customers hold keys of control through tools and features to secure their application.

8.3. Security Standards in AWS

To protect the confidentiality, integrity, and availability of its customers' systems and data, AWS has published "Overview of Security Processes" document to enable customers to build a wide range of applications without worries about any security

concerns. This document is intended to answer questions such as, “How does AWS help me protect my data?” [Osp13].

To achieve security, AWS shared responsibility with the customer. AWS adopt what is called a “shared responsibility model”, in which AWS manages the host operating system and virtualization layer down to the physical security of the infrastructure, while customer holds responsibility and management of the guest operating system, software updates and security patches. User also holds responsibility of choosing the services they need, how to integrate those services, what laws and regulations they to comply with, and what security technologies to use such as firewalls, encryption, and intrusion detection/prevention systems.

The AWS infrastructure includes the facilities, network, and hardware as well as some operational software (e.g., host OS, virtualization software, etc.). AWS manages the cloud infrastructure that customers use to provide a variety of basic computing resources such as processing and storage.

AWS hasn’t designed any specific standards of its own; instead, AWS designed and manages its infrastructure according to security best practices as well as some security compliance standards [Ras13], including:

- ***SOC 1/SSAE 16/ISAE 3402*** (formerly known as SAS 70 Type II): Is a written documentation of the internal controls that are likely to be relevant to an audit of a customer’s financial statements. AWS conducted this report in accordance with the Statement on Standards for Attestation Engagements No. 16 (SSAE 16) and

the International Standards for Assurance Engagements No. 3402 (ISAE 3402) professional standard [Soc13a].

- ***SOC 2***: Same as SOC 1, SOC 2 is used to evaluate controls. It is a confirmation report that expands the evaluation of controls to comply with the criteria set forth by the American Institute of Certified Public Accountants (AICPA) Trust Services Principles. These principles define leading practice controls relevant to security, availability, processing integrity, confidentiality, and privacy [Soc13b].
- ***The Federal Information Security Management Act (FISMA)***: all federal agencies are required to develop, document, and implement their information security system with data and infrastructure according to the National Institute of Standards and Technology Special Publication 800-53, Revision 3 standard. AWS are required to document the management, operational, and technical processes used to secure the physical and virtual infrastructure, as well as the third-party audit of the established processes and controls [Fis13].
- ***The DoD Information Assurance Certification and Accreditation Process (DIACAP)***: is the United States Department of Defense (DoD) process to ensure that risk management is applied on information systems. To maintain information assurance DIACAP defines a wide formal and standard set of activities, general tasks and a management structure process for the certification and accreditation (C&A) of any DoD information system [Dia08].

- ***The Federal Risk and Authorization Management Program (FedRAMP)***: is a government program used to regulate security assessment, authorization, and monitoring for cloud services. AWS drives a FedRAMP compliance program and is currently working in achieving compliance with FedRAMP requirements [Fed13].
- ***The Payment Card Industry, Data Security Standards (PCI DSS Level 1)***: The objective of the Payment Card Industry (PCI) Security Standards is to protect cardholder data. Financial institutions and other service providers can now run their applications on AWS PCI-compliant technology infrastructure for processing, storing, and transmitting credit card information in the cloud [Pci13].
- ***The Federal Information Processing Standard Publication (FIPS 140-2)***: is a US government security standard that specifies the security requirements for cryptographic modules protecting sensitive information [Src01].

In addition, AWS platform allows customers to deploy solutions that meet several specific industry standards, including:

- ***Health Insurance Portability and Accountability Act (HIPAA)***: The objective of HIPAA Privacy regulations is to ensure the confidentiality and security of protected health information (PHI) when it is transmitted, received, handled, or shared. AWS provides the security controls customers can use to help to secure electronic health records.
- ***Motion Picture Association of America (MPAA)***: has established a set of best practices to store, process and deliver protected media and content in a secure

manner. Media companies deploy these best practices as a way to evaluate risk and audit their content and infrastructure. AWS has demonstrated MPAA best practices and the AWS infrastructure is compliant with all applicable MPAA infrastructure controls.

Finally, to document and indicate what security controls exist in AWS Infrastructure as a Service; AWS has completed the Consensus Assessments Initiative Questionnaire (CAIQ) published by the cloud Security Alliance (CSA). The questionnaire provides a set of over 140 questions a cloud consumer and cloud auditor may wish to ask of a cloud provider.

8.4. Evaluating Security Aspects in AWS

In many companies, compliance drives security. Three years ago, AWS didn't have any certifications it could show customers to prove its carefulness, other than a limited SAS 70. Nowadays, it holds SOC 1 and SOC 2, ISO 27001, PCI-DSS, and many global third party attestations that customers can depend on and feel confident to use. AWS has made to secure its infrastructure and went through numerous third-party audits, internal audits and risk assessments to ensure to its customers that they can meet any necessary industry or government requirements.

Even though AWS adheres to a number of internationally recognized standards and protocols for data protection, privacy, and security, and did a good job of securing its infrastructure and enabling its customers to meet compliance mandates, it is fair to say

that this cloud provider simply can't get the whole credit without customers securing their part.

8.4.1. Shared Responsibility Model

AWS concept of “shared responsibility” is vague, and it doesn’t express the amount of responsibility AWS is willing to accept for security. It is kind of AWS saying “We're providing this platform for your convenience to use, but it’s your responsibility to secure what is inside”. Definitely this is not a 'shared responsibility', since customers put their trust in AWS, which, based on its documentation, tries to minimize as much security responsibility as it can.

AWS depends more on the users to choose among security features and while reading the documentation, it is noticeable the word “You” in direction to customers to take actions rather than “We” can do that for you, and you can feel that AWS is keeping itself far from any responsibility of providing full security and privacy as promised. Not only that, AWS also states that all the standard security practices pre-cloud era like adopting good coding practices, isolating sensitive data are still applicable and should be implemented.

8.4.2. Customer Awareness

When it comes to customer’s awareness of security, it is a different story. Many AWS customers don't understand that the AWS relies on a shared security model, in which, AWS is in charge of managing some specific responsibilities related to the

underlying security of the environment, but each customer must secure her own platform instances, applications and data. In other words, AWS is responsible for the physical hardware, the infrastructure, the virtualization infrastructure, and the data centers themselves. The customers are responsible for everything on top of that.

The issue is even more critical for large enterprise companies, since while many end users don't care that much about security past a certain stage, larger enterprise class customers expect and ask for a level of detail on security processes and won't move forward without that detail.

For example, if our travel agency Ajiad built its cardholder data environment on AWS PCI-validated infrastructure, the agency should secure its systems as it would in an on-premises environment, they need to harden operating systems, implement firewall rules and monitor their network traffic. Not securing its part, this agency is just putting its business out there and basically exposed.

To help its customers avoid those kinds of mistakes, AWS created documentation [Ras13] that details each of the controls, and whether responsibilities lie with AWS or the customer.

As we explained before, AWS was clear enough to define Amazon's and a customer's areas of responsibility. But what is Amazon's liability in the event if an attacker breaches or compromises its security processes? Or how Amazon would recompense customers affected by such a breach? AWS lacks fundamental details on accountability. According to [Gar12], Amazon SLAs are useless [But12].

Let's take our travel agency "Ajiad" as an example, this agency is committed to meet compliance mandates while using AWS. Whether the agency is adhering to the limited AWS certifications and compliance program or they are using other services, most of the job is on their side, they need to take care of backup and recovery processes and its baseline security policies and procedures and maintaining the infrastructure through patches and updates, which will finally result in creating a new reference document that will help them continually auditing their services.

For Ajiad, and until AWS comes up with solutions for such tasks. It is highly recommended to start with the considerable industry guidance on cloud security, mainly the National Institute of Standards and Technology's advice on security and privacy in public Cloud Computing and the CSA guidance in Cloud Computing.

8.4.3. Implementation

From an implementation point of view, the documentation "Overview of Security Process" [Osp13] is surprisingly lacking of content. There's not much in depth technical discussion, i.e., there is little real information on how AWS does what it claims it's doing.

For instance, in the security center, AWS explains key rotation, a technique of changing the customers access credentials (e.g., an AWS Access Key ID or X.509 Certificate). It is really helpful and could be used against misuse by unauthorized users. So, if we ask how Amazon performs key rotation when it's requested by a user? Sadly, there is totally no information to answer that, and if we don't understand how AWS perform the task, we can't judge how secure it is and therefore can't trust it.

8.4.4. Protection against Attacks

We can point to other obvious gaps in understanding in other sections. For example; AWS uses SSL to protect against man-in-the-middle (MITM) attacks between a user and AWS. AWS explained that by having all of the AWS APIs available via SSL protected endpoints that provide server authentication, but customers are encouraged to use SSL for all of their connections with AWS! This is wrong; SSL provides server-side authentication, but not client-side authentication. As a result, stating that SSL endpoints prevented MITM was wrong. An attacker on a compromised machine, or a user transferring data through insecure channels, is still susceptible to threats no matter what AWS does.

8.4.5. Operating Systems Protection

Other concerns revolve around how AWS secure the host operating systems, and whether those security configurations match enterprise policy needs. The same question is valid for guest VMs. Amazon states that ring 0 access (where a computer instruction can directly manipulate hardware) is restricted for virtual instances, but we need to know how. It is also unclear how AWS uses its firewalls. It says that VMs ports are closed to outside traffic until opened by the user, but what technique is AWS using to accomplish that?

8.5. Web Services Security Standards in AWS

AWS support XML-based web services and REST web services. Since the core objective of this thesis is to model and analyze security standards for web services and Cloud Computing, we are focusing on XML-based web services and their relevant web services security standards. Readers have to keep in mind that those web services security standards do not apply to REST web services.

While reviewing AWS documentations, we have noticed that Amazon barely mention web services security standards as solution to be considered. Actually, it only does this in the “Network Security” section of the documentation titled “Amazon Web Services: Overview of Security Process”. Specifically, for transmission protection, where users willing to use API with SOAP messages must secure those messages using the WS-Security standard BinarySecuritytoken profile, consisting of an X.509 certificate with an RSA public key.

We spotted some areas where web services standards could solve some AWS limitations. AWS can add more to its security process by adopting or encouraging users to use web services security standards.

For example, AWS uses its “Acceptable Use Policy” with users. This will limit the number of customers willing to use AWS, since some of them will basically disagree or dislike any conditions enforced by AWS. Using WS-Policy will give AWS more alternatives to be flexible with its assertions, and users then can use the assertion that fits their needs.

For federated users, AWS depends on external services to federate users with temporary security credentials. AWS Identity and Access Management (AWS IAM) enables the customer to grant any user temporary access to AWS resources by using security credentials that are valid only for a limited amount of time. We envision WS-Federation as adding more on federating users. WS-Federation in a secure manner will ease the process of mapping identities, reduce the cost and duplications of effort of identity, increase interoperability and achieve privacy.

For encrypting data, AWS stated that encryption of sensitive data is generally a good security practice, and AWS encourages customers to encrypt their sensitive data via an algorithm consistent with their security policy. This is a hard task for customers, especially those with no clue about securing data and applications. We will show in the next chapter” How Cloud Computing and WS security standards fit together” that we can use web services security standards such as WS-Security, WS-XML Encryption, WS-Digital Signature can be used to facilitate the encryption process within a cloud environment.

8.6. Security Flaws in Amazon Web Services

AWS have dominated a huge share of the Cloud Computing market with their ability to support from enterprise applications and big data projects to social games and mobile apps. AWS EC2 was once considered almost invincible. However, major crashes and issues with downtime problems (e.g., the outage in July 2012 that took down Netflix,

Pinterest, Instagram, and a number of other services [Kos12]) have changed the game rules for Amazon.

To evaluate AWS security and reliability claims, a group of security researchers from the European graduate technology school “Eurecom”, have published their findings about a series of security and privacy holes in the thousands of virtual machine images Amazon offers for AWS and EC2 client rental [Bal12].

The mean was to randomly choose 5,000 virtual machines from Amazon Web Services. The virtual machines were from Windows and Linux based servers equipped with a variety of web hosting and web applications software packages. AWS advertises their VMs as being “web ready,” implying that there is no needed work from the customer side to setup up basic software frameworks, such as PHP, MySQL, or Apache.

The researchers rented each VM for a short period of time, and then they configured a sequence of automated scans to search the VMs for any software vulnerabilities and exploits, common malware, and possible confidentiality holes. The results were been manually evaluated and confirmed.

The researchers were also concerned about the user account structure; they saved administrative credentials on a thousand randomly selected virtual machines, and then performed data recovery scans in order to measure the effectiveness of Amazon’s data management procedures.

Their findings included:

- ***OS Vulnerabilities:*** Surprisingly, 98% of Windows VMs and 58% of Linux VMs came pre-installed with un-patched, un-updated software with known security vulnerabilities. In particular, Windows VMs averaged 46 vulnerabilities per machine, while Linux VMs averaged 11.
- ***User Accounts:*** Over 20% of all AWS virtual machines had outdated administrator accounts or saved administrative credentials were still present on them even though the control was passed over to AWS/EC2 clients. Some of those accounts belonged to AWS administrators whose accounts according to AWS policy should have been revoked.
- ***Malware Threats:*** Two of the VM machines had potential malware infections, researchers were able to confirm Trojans that led to potential remote exploits. The researchers noted that they used the free available “ClamAV” software for the automated scanning portion, which leaves open the possibility that more serious infections might present on some Amazon virtual machines.
- ***Data Management:*** both AWS administrators and clients appeared to rarely follow data management best practices. Researchers were able to perform data recovery scans and recover deleted user files on 98% of AWS virtual machines.

Eurecom’s findings were preceded by similar ones from a group of German researchers from the Horst Goertz Institute (HGI) of the Ruhr-University Bochum (RUB). Those researchers demonstrated an account hijacking attack against Amazon Web Services (AWS) that they believe affects other Cloud Computing products as well [Hic11].

They used a technique known as XML signature wrapping or XML rewriting, known since 2005, which takes advantage of the weakness in the way web services validate signed requests. The flaw was found in the WS-Security standard and it gives the attacker the ability to trick servers to authorize digitally-signed SOAP messages that have been altered.

Using the same technique, the researchers were able to create and delete new images on the customer's EC2 instance and perform other administrative tasks like obtaining unauthorized access to an AWS account.

In addition, a separate cross-site scripting (XSS) vulnerability in Amazon's store allowed the team to take over an AWS session, which led to have free access to all customer data, including authentication data, tokens, and even plain text passwords. Amazon claimed that the identified security holes have been completely patched.

8.7. Summary

In this chapter, we looked at Amazon Web Services (abbreviated AWS), which is a collection of remote computing services (also called web services) that together make up a Cloud Computing platform, offered over the Internet.

The ultimate goal for AWS is to ensure the confidentiality, integrity, and availability of customers' data, in order to gain their trust. AWS adopts the "shared responsibility model" where AWS handles the infrastructure security while the customer is responsible for the OS and application level. From its side, AWS builds services in

accordance with security best practices, and offers different security features. But it's the user's responsibility to choose the one to use from those best practices and features to build a secure application environment.

To address Cloud customers concerns about security, AWS offer its vision of security best practices and guidelines on how to secure cloud application in the AWS environment. We summarized those best practices.

AWS also has published "Overview of Security Processes" document to give customers the chance to build secure applications. AWS hasn't designed any specific standards of its own, instead, AWS designed and manages its infrastructure according to security best practices as well as some security compliance standards. For the purpose of competing in cloud market, AWS expands its collection of security standards to include SOC 1 and SOC 2, ISO 27001, PCI-DSS, and many global third party attestations. We took a look at those standards and highlighted their functionalities.

While reviewing AWS documentations, we have noticed that Amazon barely mentions web services security standards as solutions to be considered. We highlighted some areas where web services standards could solve some AWS limitations. We identified other spots where AWS can improve its security process by adopting or encouraging users to use web services security standards. We give examples for WS-Security, WS-policy, WS-Federation, encryption, and digital signature.

AWS have dominated a huge share of Cloud Computing market; however, major crashes and issues with downtime problems have changed the game rules for Amazon:

there is more work to do. A group of security researchers from “Eurecom” published their findings about a series of security and privacy holes in the thousands of virtual machine images Amazon offers for AWS and EC2 client rental, we briefly summarized their results as OS vulnerabilities, user accounts, malware threats, and data management.

9. CONTRIBUTION OF WEB SERVICES SECURITY STANDARDS FOR SECURING THE CLOUD

Two major Cloud-standard organizations CSA and NIST reached different approaches of securing the Cloud. Their work mainly identifies Cloud key security issues, Cloud top threats, Cloud security and privacy issues, and critical areas of focus in Cloud Computing. CSA and NIST approaches are good, but miss the quality of services offered by web services standards, such as security, reliability, usability, testability, and portability. We expand here their work and add one more dimension of web services security standards. Our addition presents benefits and advantages of adopting web services security standards in securing Cloud Computing.

9.1. Matching Web Services Standards with Cloud Key Security Guidance

To address Cloud issues in this chapter, we are referencing the security guidance published by NIST and CSA that reflect the most recent approaches of identifying Cloud Computing security threats and issues. We will list their main security issues of Cloud Computing and match them with web services security standards

For all of the following security guidance, we match each security guidance input with a web service security standard that can add to it. No check mark “✓” means web

services security standards have tiny impact or no impact at all on improving that raw input.

9.1.1. CSA-Security Guidance for Critical Areas of Focus in Cloud Computing

Cloud Security Alliance (CSA) has published the “Security Guidance for Critical Areas of Focus in Cloud Computing Version 3.0” [Csa11]. This security guide outlines key issues and provides advice for both Cloud Computing customers and providers within 15 strategic domains. Their interest is in the following areas: Cloud Architecture, Governing in the Cloud, and Operating in the Cloud.

The new issue introduced into this version with respect to versions 1.0 and 2.0 is Security as a Service (SecaaS). SecaaS is aimed to secure systems and data in the Cloud as well as hybrid and traditional enterprise networks via Cloud-based services. These systems may be in the Cloud or more traditionally hosted within the customer’s premises. One of the milestones of the maturity of Cloud as a platform for business operations is the adoption of (SecaaS) on a global scale and the recognition of how security can be enhanced.

Table 6: Matching Web Services with CSA-Critical Areas of Focus in Cloud Computing

	WS Security Standards CSA-Critical Areas of Focus In Cloud Computing	SAML	XACML	XML Encryption	XML Digital Signature	WS-Security	WS-Policy	WS-Trust	WS-SecureConversation	WS-Federation
1	Governance and Enterprise Risk Management		✓				✓			
2	Contracts and Electronic Discovery		✓				✓			
3	Compliance and Audit						✓			
4	Info. Management and Data Security			✓	✓	✓	✓			
5	Portability and Interoperability	✓	✓			✓	✓	✓	✓	✓
6	Traditional Security, Business Continuity and Disaster Recovery		✓				✓			
7	Data Center Operations		✓				✓			
8	Incident Response, Notification and Remediation						✓			
9	Applications Security	✓	✓	✓	✓	✓	✓	✓	✓	✓
10	Encryption and Key Management			✓	✓					✓
11	Identity and Access Management	✓	✓	✓	✓	✓	✓	✓	✓	✓
12	Virtualization			✓			✓			
13	Security as a Service	✓	✓	✓	✓	✓	✓	✓	✓	✓

9.1.2. CSA-The Notorious Nine: Cloud Computing Top Threats

On the same track, the CSA has identified "The Notorious Nine," the top nine Cloud Computing threats for 2013 [Csa13]. The report reflects the current consensus among industry experts surveyed by CSA, focusing on threats specifically related to the shared, on-demand nature of Cloud Computing. The CSA report is meant to give Cloud service providers and their customers a snapshot of what experts see as the greatest dangers to storing data and conducting business with customers in the Cloud.

Table 7: Matching Web Services with CSA Top Threats in Cloud Computing

	WS Security Standards									
	CSA-Top Threats in Cloud Computing	SAML	XACML	XML Encryption	XML Digital Signature	WS-Security	WS-Policy	WS-Trust	WS-SecureConversation	WS-Federation
1	Data Breaches	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	Data Loss	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	Account Hijacking		✓	✓	✓	✓	✓			✓
4	Insecure APIs	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	Denial of Service	✓	✓	✓	✓	✓	✓	✓	✓	✓
6	Malicious Insiders	✓	✓	✓	✓	✓	✓	✓	✓	✓
7	Abuse and Nefarious Use									
8	Insufficient Due Diligence		✓				✓			
9	Shared Technology Issues			✓	✓	✓	✓			

9.1.3. NIST- Guidelines on Security and Privacy in Public Cloud Computing (SP 800-144)

The Guidelines provide an overview of the security and privacy challenges pertinent to public Cloud Computing, and identify considerations for organizations outsourcing data, applications and infrastructure to a public Cloud environment. The Guidelines are intended for use by federal agencies. Use in nongovernmental settings is voluntary [Jan11].

Table 8: Matching Web Services Security Standards with NIST- Guidelines on. Security and Privacy

	WS Security Standards									
	NIST- Guidelines on Security and Privacy in Public Cloud Computing	SAML	XACML	XML Encryption	XML Digital Signature	WS-Security	WS-Policy	WS-Trust	WS-SecureConversation	WS-Federation
1	Governance		✓				✓			
2	Compliance		✓				✓			
3	Trust		✓			✓	✓	✓		
4	Architecture									
5	Identity and Access Management	✓	✓	✓	✓	✓	✓	✓	✓	✓
6	Software Isolation									
7	Data Protection	✓	✓	✓	✓	✓	✓	✓	✓	✓
8	Availability						✓	✓		
9	Incident Response									

9.1.4. NIST-Cloud Computing Synopsis and Recommendations

NIST released Special Publication (SP) 800-146, Cloud Computing Synopsis and Recommendations [Bad12]. It gives an overview of the security and privacy challenges facing Cloud Computing and corresponding recommendations that organizations should follow to when planning, reviewing or negotiating a Cloud Computing service with providers. Weighing in at 81 pages, this document is detailing 23 “open issues” for Cloud that requires further analysis. These issues were grouped into five categories which are: Computing Performance, Cloud Reliability, Economic Goals, Compliance, and Information Security.

Table 9: Matching Web Services Security Standards with NIST-Cloud Computing Synopsis and Recommendations

	WS Security Standards									
	NIST-Cloud Computing Synopsis and Recommendations	SAML	XACML	XML Encryption	XML Digital Signature	WS-Security	WS-Policy	WS-Trust	WS-SecureConversation	WS-Federation
	Computing Performance									
1	Latency						✓			
2	Off-Line Data Synchronization						✓			
3	Scalable Programming						✓			
4	Data Storage Management	✓					✓			✓

	Cloud Reliability									
5	Network Dependence		✓				✓			
6	Cloud Provider Outage		✓				✓			
7	Safety-Critical Processing		✓				✓			
	Economic Goals									
8	Risk of Business Continuity		✓				✓			
9	Service Agreement Evaluation		✓				✓			
10	Portability of Workloads	✓	✓			✓	✓	✓	✓	✓
11	Interoperability between Cloud Providers	✓	✓			✓	✓	✓	✓	✓
12	Disaster Recovery		✓				✓			
	Compliance									
13	Lack of Visibility		✓				✓			
14	Physical Data Location		✓				✓			
15	Jurisdiction and Regulation		✓				✓			
16	Support for Forensics		✓				✓			
	Information Security									
17	Risk of Unintended Data Disclosure		✓	✓	✓	✓	✓	✓	✓	✓
18	Data Privacy	✓	✓	✓	✓	✓	✓	✓	✓	✓
19	System Integrity		✓				✓			
20	Multi-tenancy		✓				✓			
21	Browsers		✓				✓			
22	Hardware Support for Trust		✓				✓	✓		
23	Key Management		✓	✓	✓		✓			✓

9.1.5. Cloud Computing Quality Attributes

The following table summarizes what each web services security standard is good at once considered to be deployed in a Cloud system. Those conclusions came from our

study and analysis of the web services standards. The only web services standards we are taking into consideration are the ones specifically related to security. Some of those web services standards have been presented in the previous chapters of this dissertation while others were completed by the Secure Systems Research Group (SSRG) [Ssr13]. The future work includes expanding this approach to address all other web services security standards.

Table 10: Matching Web Services Standards with Cloud Computing Quality Attributes

	WS Security Standards									
	Cloud Computing Quality Attribute	SAML	XACML	XML Encryption	XML Digital Signature	WS-Security	WS-Policy	WS-Trust	WS-SecureConversation	WS-Federation
1	Adaptability	✓					✓			✓
2	Auditability									
3	Availability							✓		
4	Extensibility	✓				✓	✓			
5	Interoperability	✓	✓		✓	✓	✓	✓	✓	✓
6	Modifiability	✓				✓	✓			
7	Deployability						✓			
8	Performance									
9	Reliability				✓	✓				
10	Scalability	✓								✓
11	Security	✓	✓	✓	✓	✓	✓	✓	✓	✓
12	Testability									
13	Usability	✓								

A comprehensive analysis of security issues for Cloud Computing could be found in [Has13]. They have presented security issues for Cloud models: IaaS, PaaS, and SaaS, which vary depending on the model. While most surveys have discussed security issues about Clouds without making any difference between vulnerabilities and threats, this paper focused on this distinction, where they consider it important to understand these issues. They indicated that enumerating Cloud security issues is not enough; for which they made a relationship between threats and vulnerabilities, so it is easy to identify what vulnerabilities contribute to the execution of these threats and make the system more robust.

9.2. Summary

A big reason behind adopting web services security standards is their key quality attributes such as interoperability, extensibility, and modifiability. This is true for Cloud Computing. Two major Cloud-Standards organizations NIST and CSA did a good job of defining critical areas of securing Cloud Computing, but none of them took web services security standards into consideration. By adding web services security standards, we saw what area of improvements can be achieved, and how web services security standards could help to implement proposed goals for both organizations.

10. ADOPTING WEB SERVICES SECURITY STANDARDS IN CLOUD COMPUTING

We discuss now possible use of the web services security standards presented earlier in a cloud to control the use of any type of service. In addition to XML-based services, clouds use heavily REST-based services. To structure the discussion we use a style similar to a pattern template.

Before going into details, we can say that REST and SOAP technologies can be mixed and matched. REST is simple, very easy to understand and is extremely approachable, but does lack standards. In comparison, SOAP is an industry standard with a well-defined protocol and a set of well-established rules to be implemented, and it has been used in systems both big and small.

To give an example, SOAP has a set of standard specifications. WS-Security is the specification for security. It is a detailed standard providing rules for security in application implementation. Like this we have specific standards for messaging, policy, trust, federation, etc. Unlike SOAP, REST does not have dedicated standards for each of these. REST predominantly relies on HTTPS. Similarly, REST doesn't have a standard messaging system and expects users to deal with communication failures by retrying. SOAP has more controls around retry logic and thus can provide more end-to-end reliability and service guarantees.

For that reason, we are focusing only on SOAP-based web services since the enterprise software that required complex transactions, monitoring, and orchestration capabilities relied on the SOAP-based architecture and standards to realize their SOA efforts. This is true for Cloud systems since Cloud is a special case of SOA.

Unfortunately, there isn't enough standardization when it comes to REST, and to be specific from REST APIs from one cloud provider to another. The future work of this thesis will investigate how to use XML security to control REST services.

Adopting web services security standards into Cloud Computing solves some key Cloud Computing issues; the following sections summarize our findings.

10.1. Issue: Need for Policies

Cloud services need to communicate in a collaborative way to perform work. Cloud services can be composed through nesting and layering with other Cloud services. For example, a public SaaS provider can use PaaS or IaaS Cloud environments to build its services.

In order to assure reliability, availability and security, Cloud services providers need to apply policies. Without them, they will have no means to specify what quality factors they enforce and require from their users. This situation would result in all kinds of problems for the Cloud providers and customers keeping in mind they are using the internet which is an insecure and unreliable environment.

The security policies and practices of the Cloud provider might not be adequate to match customers' needs or compatible with other Cloud services. Some of the complications could be: Loss of privacy where Cloud provider does not handle sensitive information as the user policy indicates, or undetected intrusions or violations due to insufficient auditing and monitoring policies deployed by the Cloud provider.

10.1.1. Example

Ajiad is a travel agency that intends to provide online services to its customers. *Ajiad* now offers many of its everyday operations as a Cloud-based system that uses XML-based services. In the current situation, some of *Ajiad's* customers have been accessing Cloud services they are not allowed to access. The reason for that are outdated and weak services not having systematic guidelines to control their use. Some of the services are not up when needed. As a result *Ajiad* is losing money because of its reliability and security problems.

10.1.2. Solution: WS-Policy.

WS-Policy provides a set of specifications that describe the capabilities and constraints of the security (and other business) policies on intermediaries and end points (for example, required security tokens, supported encryption algorithms, and privacy rules) and how to associate policies with services and end points [Wsp07].

We can use WS-Policy to define the interaction between Cloud vendors and their consumers by using XML to advertise their policies (on security, Quality of Service, etc.) and for consumers to specify their policy requirements.

WS-Policy also provides a way to check the requests made by requestors in order to verify that they satisfy their assertions and their conditions before interacting with the web service. Policies can be defined for security, reliability or other business constraints. For example, Cloud services can be protected against unauthorized access by having policies that provide conditions in order to use them. Consumers willing to use Cloud services are required to comply with its policy.

For example, Cloud vendors can attach their policy to their offers, clients wanting to transact must comply with its conditions (e.g. signing, encryption, timestamp, and username) as specified in the policy. More details about how to specifically define policies can be found in Chapter 4. In general, any entity in a client-server system may expose a policy to convey conditions under which it provides service. Satisfying assertions in the policy usually results in behavior that reflects these conditions. For example, if two entities - Cloud vendor and customer - expose their policies, a customer might use the policy of the provider to decide whether or not to use the service. To satisfy more customers' needs, a Cloud vendor can expose alternative policies, and a customer may choose any alternative since each is a valid configuration for interaction with the service, but only one of them.

Implementing the WS-Policy in the Cloud presents the following advantages:

- *Policy protection.* It is possible to define policies to protect the policies themselves.

- *Policy integrity*: Using the appropriate signing mechanism will protect the policy assertions from tampering, e.g. Customers can discard a policy which is not signed by the Cloud provider or when not presented with sufficient credentials.
- *Data security of web services*: It is possible to secure the data of Cloud web services, since we can use policies from other web services standards such as WS-Security [Ibm04], XML Digital Signature [Xds08] and WS-Metadata Exchange [Wme04].
- *Guaranteed message exchange*: Cloud providers the ability to avoid older or weaker policy alternatives and by giving the requestor the ability to verify the policy provider.
- *Availability*: The WS-Policy standards mitigates the chance of denial of service threats by enforcing the policy implementers to use a model with defaults on the policy alternatives, number of assertions in an alternative, and depth of nested policy expressions.

10.1.3. Example Resolved

Ajiad now defined systematic rules to specify the way its Cloud services should be accessed in terms of who, when, and in what, as well as conditions *Ajiad's* new web-based system now has more control over its services by applying prerequisite conditions and security constraints through policies. So, in order to use any service, all customers are required to comply with its policy conditions and satisfy its requirements (for example, by using the required security token types specified by the policy) and agree with its terms before using the web service.

10.2. Issue: Authorization and Access Control

The typical Cloud environment is heterogeneous includes different vendors, different platforms as well as a variety of end users. At the same time, this environment is opened to a wide variety of partners, customers or mobile employees, which introduce a new variety of security threats. The Cloud vendors must protect their information assets from attacks. Their information assets typically include hardware infrastructure, OS platforms, and services, which come in a variety of technologies, components, and data.

Cloud providers need to define authorization, which are high level guidelines that specify who is allowed to do what to resources to protect their assets [Fer12]. Not only that, but also enforce these policies by security mechanisms. In a nested environment like the Cloud, the policies may be issued by different actors making their management challenging.

10.2.1. Example

The virtual travel agency “*Ajiad*” provides travel services through to its customers. Their Cloud can be accessed by customers who book reservations and flights to the company for buying or selling tickets. *Ajiad* have business relations with other partners to guarantee a high satisfaction. *Ajiad* can carry out the orders of the customers by itself or send requests to other partners for completion.

All those transactions are regulated by security policies within “*Ajiad*”. For example, the billing department can have the rule «only valid registered customers are allowed for booking», the company security policy can state that «only partners with

“VIP” privilege can place “VIP” reservations» and that «only partners with “financial institution” state can process financial transactions», etc.

All these policies are enforced by different components of the computer system of the company (email, online, internal software, etc.). This approach introduces many obstacles: The policies are described in possibly different syntaxes and it is difficult to have a global view of what policies apply to a specific case. Moreover, overlapping between policies is possible and there is no way to combine them in a clear way. In other words, this approach could be error-prone and difficult to manage.

10.2.2. Solution: XACML

XACML (eXtensible Access Control Markup Language) has been defined by OASIS and it includes languages for expressing authorization rules and for access decision (enforcement of the rules). We need to address Cloud users’ privileges and preserve control over access to resources. SAML by itself is not enough to handle ID and access management in Cloud, XACML standard can be used for that purpose. The XACML standard describes an XML-based language for describing authorization policies and making access control decisions. The difference between XACML and SAML is that, while XACML focuses on the mechanism for arriving at authorization decisions, SAML focuses on the means for transmitting authentication and authorization decisions between Cloud entities.

Some Cloud providers already have XACML in place. XACML neither specify how user credentials are validated nor define protocols or transport mechanisms. The

XACML concept assumes that when an entity attempt to access a resource, a Policy Enforcement Point (PEP), responsible for protecting access to resources, sends a request that describes the attempted access to a Policy Decision Point (PDP) to evaluate that request against pre-defined policies and attributes. Once evaluated, The PDP returns an authorization decision for the PEP again to enforce. Patterns for XACML authorization and access control can be found in [Del05].

10.2.3. Example Resolved

Ajiad can implement XACML authorization rules to centralize a wide range of policies and rules. Those can be easily managed; we can resolve conflicts by evaluating access requests using rights combining algorithms. *Ajiad* also can use XACML Access Control to centralize the decisions of accesses to resources in the company. Consequently, partners and employees do not need to care about access control decisions anymore, since every access request or response is in the XACML format.

Even though XML is a verbose language, and it could affect the performance of the protected system (by taking a longer processing time); implementing the XACML Policy Language in Cloud presents the following advantages:

- Cloud providers can use XML to define policies to control access. Policies and rules can be easily combined, thus making the Cloud system less complex and more secure.
- XACML can support the access matrix, RBAC or multilevel models for access control.

- The access decisions can be requested in a standard format, an access decision becomes independent from its enforcement. A broad variety of enforcement mechanisms could be supported and can evolve separately from the PolicyDecisionPoint.

10.3. Issue: Data Protection during Transmission

Typically, data stored in a Cloud environment resides in a shared environment place together with data from other customers. Priority no.1 for users should be how to prevent data from being accessed by unauthorized users or even malicious processes or rogue virtual machines running in the Cloud.

Therefore, customers must understand the ways of controlling access to the data and keeping the data secure. The same is true for data transmitted within or between Clouds. Data must be secured in all stages, while at rest, in transit, and in use, and access to the data must be controlled. For that purpose, standards for communications protocols and public key certificates can be used to protect data transfers using cryptography. The usage of encryption as a technique to secure data guarantees the confidentiality of data and helps to detect any corruption in data.

Cryptography can be implemented in SaaS, PaaS, and IaaS models as well taking into consideration that capabilities vary significantly across Cloud models, and cryptographic mechanisms may not be feasible for PaaS and SaaS environments [Bar05]. Some researchers suggested offering cryptography as a service in Cloud Computing environment to alleviate this limitation [Ide12].

The security degree of a system employing cryptography depends on implementing proper control of central keys and key management components. Currently, Cloud consumers hold the responsibility for cryptographic key management. NIST's Cryptographic Toolkit selects cryptographic security components and functionality for protecting their data, communications, and operations [Csd13].

10.3.1. Example

Adnan is a manager in the purchasing department in *Ajiad*. He regularly sends reservations bills to a bank (another Cloud partner), where Zarifah in the billing department processes these bills. A reservation bill contains sensitive data such as user information and credit card numbers, so it is important to keep it secret. An attacker can intercept their messages and may try to read them to get the confidential information. From her side, Zarifah wants to be certain that the bill was created by Adnan so she can credit the money to *Ajiad* account.

10.3.2. Solution: WS-Security.

XML Encryption and XML Signature are two of the basic standards in securing web services, and these standards are used by other emerging standards such as WS-Security. The XML Encryption standard defines the process of encrypting and decrypting all of an XML message, part of an XML message, or even an external resource. [Has09] explains more the technical part of those standards.

Some Cloud vendors let the user choose the encryption method that fits their needs, for example, AWS gives customers options to choose from. Any open source or

commercial encryption software can be used to encrypt the data before storing it as Amazon S3 objects and decrypt it after download.

WS-Security defines how to embed XML encryption and XML signature into XML documents. It also defines how to embed security tokens such as Kerberos Tickets and X.509 which provide message authentication. WS-Security does not define new security mechanisms but leverages existing security technologies such as encryption and digital signature [Wss06].

10.3.3. Example Resolved

Adnan now encrypts the reservation bills she sends to Zarifah. The reservation bill sensitive data is now unreadable to the attackers. An attacker can try to apply to it all possible keys but if the algorithm has been well chosen and implemented, she cannot read the confidential information.

Ajiad and its bank partner agree on the use of a digital signature algorithm. The bank has access to *Ajiad's* public key. *Ajiad* can then send a signed message to the bank. Once received, the bank verifies whether the signature is valid using *Ajiad's* public key and the agreed signature algorithm. If the signature is valid, the Bank can be confident that the message was created by *Ajiad*.

Implementing XML encryption and XML signature brings the following pros:

- *Multiple signatures*: XML digital signature allows multiple signers. We can sign different parts of a message with different signatures. This allows a set of principals to write portions of a document and sign them.
- *Non-repudiation*: When a signature is validated using a principal's public key, the sender cannot deny that he created and sent the message.
- *Overhead*: The available algorithms that can be used for digital signatures do not require very large amounts of computational power and do not take large amounts of time.
- *Validation of signature*: An XML signature is an XML element that is embedded in the message. The XML signature is composed of several XML elements that include information such as the value of the signature, the key that will be used to verify the signature, and algorithms used to compute the signature. This standard format helps XML parsers to better understand signature elements during the validation process.
- *Verification of signatures*: Because a principal's private key is used to sign the message, the signature can be validated using its public key, which proves that the sender created and sent the message.

10.4. Issue: Trust

Since customers lack control of Cloud resources, they are not in a good position to utilize technical mechanisms in order to protect their data against unauthorized access or secondary usage or other forms of misuse. Instead, they must trust that the SP will protect their information.

Trust is a complex concept for which there is no entirely accepted definition. Trust is based on security and other policies to enable requesting and obtaining credentials within different trust domains. Trust can be established in many ways: security may be one of these (although security, on its own, does not necessarily imply trust [Ost01]). On the other side, an example of increasing security to increase trust comes from people being more willing to do online shopping if they are assured their personal data and credit card numbers are cryptographically protected.

Establishing a level of trust about a Cloud service is dependent how much the Cloud provider is able to satisfy the customers' requirements, and what are security controls necessary to protect their data and applications, and also the evidence provided about the effectiveness of those controls [Jts10]. Now, if verifying the correct functioning of a subsystem and the effectiveness of security controls is not feasible, third-party audits may be used to establish a level of trust.

Trust between the Service provider and the customer is a major issue and has received strong attention by companies. There is no way for the users to guarantee whether the management of the Service is trustworthy, and whether there is any risk of insider attacks. Nowadays, the Service Level Agreement (SLA) is the only legal document between the customer and service provider. It contains what the service provider considers it can do in a reasonable way. Yet, at present, there is no agreed content for the SLA, and as a consequence, there might be undocumented services in the SLA that the customer has no clue about. Even worse, most SPs will not guarantee security.

Moving data and applications to a Cloud environment managed by a Cloud provider expands the circle of insiders to include not only the Cloud provider's staff, but also other customers using that service, thus increasing risk. The insider security threat is a well-known issue that applies to Cloud services as well [Fer12].

Cloud services providers might depend on other Cloud providers to deliver services for customers. For those types of nested Cloud businesses, trust relationships in the Cloud chain may be weak, but needed in order that a service can be provided. This scenario combined with the globalized open nature of Cloud opens the door for significant business risk due to loss of control in passing sensitive data between Cloud providers.

Because SPs use other SPs, not known to the SCs, trust cannot be transitive. Actually, due to a lack of transparency, customers may have no idea at all about the identity of the Cloud providers in this nested Cloud chain. For example, 'pay per usage' models may be established on weak trust relationships, involving third-parties vendors with weak data security mechanisms and vague practices that easily expose customers to malicious attacks. In some scenarios, where saving time and delivering quick services, new third-parties could be added to the chain for which customers have no chance to know them, much less to check about their practices, reputation and trustworthiness.

Ultimately, usage of the Cloud is a question of tradeoffs between security, privacy, compliance, costs and benefits. Trust is a key to adoption of SaaS, and

transparency is an important mechanism. Furthermore, trust mechanisms need to be propagated right along the chain of service provision.

10.4.1. Example

The *Ajiad* travel agency offers its travel services through several different business portals to provide travel tickets, hotel and car rental services to its customers. *Ajiad* needs to establish trust relationships with its partners through these portals.

The *Ajiad* supports different business relationships and needs to be able to determine which travel services to invoke for which partner and for which customer. Without a well-defined structure, *Ajiad* will not be able to know if a partner should be trusted or not, or to automate the trust relationships quickly and securely with its partners, which may lead to losing a valuable business goal of offering integrated travel services. Some customers or partners may not be reliable and not pay or provide services, so we have to make sure we deal only with trusted entities.

10.4.2. Solution: WS-Trust

Cloud systems interact with customers and other Cloud services. Sometimes users and Cloud systems are not known to each other and a trust relationship must be established before any interaction between the participants. This relationship can be defined by exchanging security tokens such as certificates or other proofs of identity or attributes. WS-Trust is a standard to support the establishment of trust relationships between web services [Wst09]. Trust depends also on reputation but this is an aspect not included in the standard.

Using web services requires that we exchange credentials to define the rights of each participant. This exchange is based on trust and builds further trust. Trust is based on security and other policies to enable requesting and obtaining credentials within different trust domains. Both parties need to determine if they can "trust" the asserted credentials of the other party. The goal of the WS-Trust standard is to enable applications to construct trusted message exchanges. This trust is realized through the exchange and brokering of security tokens.

The motivation toward WS-Trust is supported by the fact that there are different formats for security tokens (e.g. X.509 certificates, Kerberos tickets, SAML assertions, XACML policies, etc.), and it's unlikely to expect that an endpoint will understand each of these options. Additionally, there is no guarantee that there will be an intersection between the sets of supported security token formats of different actors who are willing to exchange messages using the WS-Security standard [Mad03]. The WS-Trust has been explained more in [Aja10a].

WS-Trust standard can be implemented in Cloud-based systems and brings the following advantages:

- *Interoperability*: The credentials can be translated when necessary.
- *Knowledge*: A token encapsulates some knowledge about a potential service or requestor. The presence of valid tokens can give us some assurance of their trust level.

- *Message security and authenticity*: All the messages exchanged between the involved parties are signed (XML Digital Signature) and encrypted (XML Encryption).
- *Policy consideration*: The Cloud provider can implement a trust engine that applies the policies when validating claims.
- *Policy protection*: By extending the WS-Security mechanisms, we can handle security issues such as security tokens (the possibility of a token substitution attack), signing (where all private elements should be included in the scope of signature and that this signature must include a timestamp).
- *Time validity*: We can specify time constraint in the parameters of a security tokens issued by STS. This constraint will specify for how long that security token is valid. Upon expiring, the security token's holder may renew or cancel it. Cloud providers have more control over security tokens sessions.
- *Trust*: With this solution, we have the choice of implementing WS-Policy framework to support trust partners by expressing and exchanging their statements of trust. The description of this expected behavior within the security space can also be expressed as a trust policy.

10.4.3. Example Resolved

Ajiad now has the ability to automate its trust relationships with its partners by managing the registration tasks for all its partners and issuing customers a unique ID's. In this case, *Ajiad* becomes a mediator between the customers and its participant partners

and plays the role of negotiator and third-party player who is looking to satisfy both sides.

Ajiad now can offer a Security Token Service for its business partners, who may find useful ways to take advantage of credit processing and other services offered by *Ajiad*, which now has new business opportunities. *Ajiad* itself can reduce non-payments and lack of integrity of its partners because it can deal only with trusted customers and partners.

10.5. Issue: ID Management

The growth in business-to-business commerce, increased mobility and the importance of persistent interactions between involved parties are some of the current Cloud market challenges. To meet these challenges, Cloud vendors are extending internal systems to external users of different categories (employees, customers and partners). A variety of users who need to interact with a variety of autonomous Cloud systems requires a careful handling of identities. Building secured and trust-based relationships among users might require sharing their identity information. Trust relationships should allow identity and policy data to be exchanged between parties independent of platform, application or infrastructure, and avoid redundant work.

The Identity management involves the use, maintenance, and protection of users' credentials. Preventing unauthorized access to those credentials in the Cloud is a must. One management issue is that the organizational identification and authentication ID management systems may not be fully extensible into a Cloud environment and

attempting to change the existing systems to support Cloud services may prove difficult [Cho09].

10.5.1. Example

Our travel agency “*Ajiad*” is a part of travel agencies consortium. The goal of this consortium is to expand a partner business and give it privileges to reach other members domains. Each domain’s employees have authorized access to other partners’ resources. Each member controls its own resources and has the final access control decision.

Without a well-defined structure of sharing identities with other parties, *Ajiad* will not be able to determine which travel services to invoke for a given customer, or determine how to allow businesses to directly provide services for customers registered at other (partner) businesses, or allow disparate security domains to broker information on identities, identity attributes and authentication. Not having this structure may lead to losing a valuable business goal of offering integrated travel services.

10.5.2. Solution: WS-Federation

A Federation describes the technology and mechanisms necessary to systemize this interconnection, and to allow different domains to use identities from different domains. Identity federation was promoted with the introduction of SOA. With identity federation providers and users of Cloud can trust and share digital identities and attributes to provide means for a single sign-on approach. For such a federation to properly work, identity and access management transactions must be protected against attacks.

For that reason, the WS-Federation standard defines mechanisms to allow different security domains (realms) to federate their identities, such that authorized access to resources managed in one domain can be provided to principals whose identities are managed in other domains [Wsf09]. Those federation mechanisms enable the decision of federating IDs to be based on the declaration (or brokering) of identity, attribute, authentication and authorization assertions between domains. Addressing all these concerns in one abstract solution will facilitate the interaction between web services. Details of WS-Federation are presented in Chapter 3.

Similar to WS-Federation, SAML allows users that have already logged into one site to access another site without logging in again. An increasing number of Cloud providers use the SAML web services standard to manage and authenticate users before granting them access to applications and data. SAML focuses on the means for transmitting authentication and authorization decisions between Cloud entities. For example, a SAML transaction can tell that a user has been authenticated by an identity provider, and she is allowed to have certain privileges. Once receiving that transaction, the service provider verifies the user credentials, and uses the information to grant the user an appropriate type of access.

Implementing the WS-Federation in Cloud systems presents the following advantages:

- *Degree of Security*: Cloud vendors can develop offline operating agreements with other service providers to agree about architecture and privacy policies. They can

use mechanisms provided by other web services specifications such as WS-Security to secure access to the policy, XML Digital Signature to authenticate sensitive information and WS-Metadata Exchange to describe what other endpoints need to know to interact with them.

- *Identity Centralization*: To reduce the cost and duplication of effort of identity management; each Cloud partner's identity is almost always managed by a trusted partner.
- *Identity Mapping*: Cloud vendors within a federation don't need to register and maintain other users' identities, and the user is spared from having to get and remember a new login in order to interact with the business. This is done through mapping trusted information about a foreign user (e.g., users from business partners) into authentication and authorization information usable by another partner's resources.
- *Interoperability*: By providing required credentials, and agreeing upon privacy policies, Cloud partners could have their own federated identity that is gradually and transparently created to be used within a federation. Web services standards—including SOAP, XML, WSDL, and UDDI—successfully enable developers to create web service solutions that are interoperable across multiple platforms, programming languages and applications.
- *Privacy*: While obtaining federated identity within a federation, partners can classify some of their attributes as private; therefore an identity provider can identify which attributes it shouldn't transmit to other parties. Which attributes are

considered private and which are not, depends on the user preferences governing the use of their own data.

10.5.3. Example Resolved

By utilizing WS-Federation mechanisms, our travel agency is able to provide access to its services without the overhead of managing other partners' users. This reduces administrative costs (because the accounts for the employees of their partners are handled by the partners themselves) and provides improved service for partners. As each partner employee account is managed by the user's employer, the accuracy of user's attributes asserted in claims is greatly improved because the partners know the most current status of their employees. This, in turn, improves security when it comes to access control decisions based on the most up-to-date user context with no worry about orphaned user accounts associated with former users of other partners.

10.6. Summary

One of the weak points of Cloud Computing is the security problem because data or information are stored remotely on a server. Web services standards give a good solution to the security problem so that Cloud Computing is easier to be accepted in the business cases which require high protection of their data. There are many web services standards that cover different domains, some of which include: message exchange, transport, security, reliability, trust, federation of identities, business process, and service publications. These are all essential elements in a Cloud Computing process. In this chapter we showed how deploying web services standards can solve some Cloud issues.

11. CONCLUSIONS AND FUTURE WORK

Cloud Computing systems are complex systems that leverage different technologies and can be deployed in different ways. While Cloud adoption is expected to speed up in the coming years, companies are still cautious about the Cloud as the right delivery medium for their services. The dominant concern is security. Since Cloud Computing is a relatively new Computing model, there is a great deal of hesitation about how to achieve security at all levels. Security concerns are spotted in different areas such as external data storage, dependency on the open “public” internet, lack of user control, multi-tenancy and integration with internal security.

For those security concerns, more and more demands for Cloud standards are put on the table. The Cloud standardization landscape is so ambiguous because there isn't a central body or forum to control the process of standardization.

The case with Cloud is similar to what happened around 2006 with web services. Too many standards to support different quality attribute. The lack of standards also makes it difficult to establish security frameworks for such heterogeneous environments and forces people for the moment to rely on common security best practice.

The main purpose of this thesis is to model and analyze security standards for web services and Cloud Computing. We first continued the work we have done before of writing patterns for web services security standards, and then we used web services

security standards to solve major Cloud issues, such as, need for policies, authorization and access control, data protection in transit, trust, and ID management.

In this work, we have provided the following contributions:

1. We presented a pattern for the WS-SecureConversation (Chapter 2) that described how a web service can authenticate requester messages, how requesters can authenticate services, and how to establish mutually authenticated security contexts.
2. We have presented a pattern for the WS-Federation (Chapter 3) that described how to manage and broker the trust relationships in a heterogeneous federated environment, including support for federated identities, sharing of attributes, and management of pseudonyms.
3. We defined what relationships (Chapter 4) exist between web services security standards. We drew a pattern diagram that showed what degree of dependency a pattern has with other patterns in details.
4. We added security to BPEL workflows of web services (Chapter 5). We considered UML activity diagrams for collaborative business processes and showed how to list the possible threats and attacks that could happen in order to define the appropriate and suitable countermeasures to stop or mitigate them.
5. The need for Cloud Computing standards (Chapter 6) aimed to solve the confusion about the need for standards in Cloud Computing. We defined what a

standard is, and we explained what makes a good standard. We listed the main factors of why do we need standards for Cloud Computing.

6. We surveyed in chapter 7 work on security standards for Cloud Computing and we classified them in groups depending on their functionalities. We also included standards that although not developed for Cloud Computing, have an impact on the use of clouds. We listed in the main issues of Cloud Computing standardization. We briefly presented some industry efforts.
7. In chapter 8, we took Amazon Web Services (AWS) as a case-study to see how a Cloud vendor looks at web services standards, and whether it takes them into consideration while offering services. We highlighted some areas where web services standards could solve some AWS limitations. We identified other spots where AWS can improve its security process by adopting or encouraging users to use web services security standards.
8. Chapter 9 selected two major Cloud-standards organizations, CSA and NIST, and examined their security guidance of securing the Cloud. We noticed that both missed the quality of attributes offered by web services standards. We expanded their work and added one more dimension of web services security standards.
9. Chapter 10 matched major Cloud security issues with solutions offered by web services security standards, some of which include: message exchange, transport, security, reliability, trust, federation of identities. Adopting web services security

standards gives a good solution to the security problem so that Cloud Computing is easier to be accepted for the businesses which require high protection of data.

The future work of this thesis includes the following:

- **Develop more patterns for web services security standards**

In Chapter 5 we presented a pattern diagram for web services security standards, we are going to develop more patterns for other web services security standards and extend the pattern diagram to include them.

- **Develop a design model for BPEL**

For BPEL, We have presented an approach that enumerates the threats to a given BPEL process. We considered UML activity diagrams for collaborative business processes and showed how to list the possible threats and attacks that could happen in order to define the appropriate and suitable countermeasures to stop or mitigate them. The use of UML activity diagrams produces a clear and more intuitive way to analyze these attacks than working directly in BPEL.

Future work for BPEL includes a design model for our example of travel agency, which will explain how to deploy web services standards such as WS-Security, WS-Policy and WS-Trust in a systematic way. Such a design model will use our threat enumeration approach to specify exactly which security mechanisms we should deploy for which classes. We intend to incorporate also this approach as part of the secure application design methodology of [Fer06a], which now starts from use cases instead of

workflows. This would allow building systems combining web services and standard components, which are necessary in some architecture. Our approach starts from use cases and class/sequence diagrams where BPEL models can be done using an approach such as the one in [Shi05].

- **Develop Patterns for Cloud Computing Standards**

We see the use of patterns as a fundamental way to implicitly apply security principles even by people having little experience. A complete catalog is a fundamental tool to design complex systems. We have proposed a development approach that applies security throughout the whole lifecycle and uses security patterns [Fer06a]. As part of this work we have produced a variety of security patterns for all the architectural levels of the system [Fer13].

In Cloud Computing, there are many standards developed by many organizations. The standardization process is ambiguous, because there isn't a central body or forum to control the process of standardization, despite the efforts made by lots of people and organizations on that direction. Our survey counted "66" standards, which are rather complex and verbose and it is not easy for designers and users to understand their key points, only "9" of those have patterns describe them.

Building Patterns for Cloud Computing standards will ease the process of implementing them in Cloud environments. It will make it easy for Cloud vendors to deploy in their model and easy for users to choose from the ones that matches their needs. The abstracted form of these Cloud standards make them applicable to challenges that

developers of cloud application face today, independent of the actual technologies and cloud services that they are using.

- **Use web services security standards to solve more Cloud challenges**

Web services security standards give a good solution to the security problem so that Cloud Computing are easier to be accepted in the business. In Chapter 10, we matched major Cloud security issues with solutions offered by web services security standards, some of which include: message exchange, security, trust, federation of identities. We are going to propose more solutions to other Cloud challenges such transport and reliability using the new patterns for web services security standards.

- **Use XML Security to control REST-based services in the Cloud**

Cloud Computing is not only about XML-based web services, Cloud use heavily REST-based services too. The future work of this thesis is investigating how to use XML security to control REST-based services.

- **Adopt web services security standards into Cloud-standards organizations approaches of securing the Cloud.**

Two major Cloud-developing organizations CSA and NIST reached different approaches of securing the Cloud. Both approaches are good, but miss the quality of attributes offered by web services standards. We expanded their work in Chapter 9 and add one more dimension of web services security standards. We are going to investigate

more approaches of securing the Cloud of other Cloud-standards organization and see how web services security standards can contribute to their work.

12. REFERENCES

- [Aes01] NIST, FIPS 197: Advanced Encryption Standard (AES), <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf> - Last accessed on Mar 05, 2013.
- [Aja10a] O. Ajaj and E.B.Fernandez, "A pattern for the WS-Trust standard of web services", Procs. of the 1st Asian Conference on Pattern Languages of Programs (AsianPLoP 2010), Tokyo, Japan, March 16-17, 2010, <http://patterns-wg.fuka.info.waseda.ac.jp/asianplop/>.
- [Aja10b] O. Ajaj, and E.B. Fernandez, "A pattern for the WS-Policy standard", Procs. of the 8th Latin American Conference on Pattern Languages of Programs (SugarLoafPLoP 2010), Salvador, Bahia, Brazil, Sept 23-26, 2010, <http://wiki.dcc.ufba.br/SugarLoafPlop>
- [Aja12] O. Ajaj, and E.B. Fernandez, "A pattern for the WS-SecureConversation of web services", Procs. of the 19th Conference on Pattern Language of Programs (PLoP 2012). Tucson, AZ October 19-21, 2012, <http://www.hillside.net/plop/2012/>
- [Aja13] O. Ajaj and E.B. Fernandez, "A pattern for the WS-Federation of web services ", under review, the 20th conference on Pattern Languages of Programs, October 23–26, 2013—Allerton Park, Monticello, IL, USA. <http://www.hillside.net/plop/2013/>
- [Apa13] Apache CXF, WS-SecureConversation, <http://cxf.apache.org/docs/ws-secureconversation.html>

- [Apg93] NIST, FIPS 181: Automated Password Generator (APG), <http://www.itl.nist.gov/fipspubs/fip181.htm> - Last accessed on Mar 05, 2013.

- [Bad12] L. Badger, T. Grance, R. Patt-Corner, and J. Voas, NIST, “Cloud Computing Synopsis and Recommendations”, <http://csrc.nist.gov/publications/nistpubs/800-146/sp800-146.pdf> - Last accessed on April 15, 2013

- [Bal12] M. Balduzzi, M., J. Zaddach, D. Balzarotti, E. Kirda, and S. Loureiro, A Security Analysis of Amazon’s Elastic Compute Cloud Service, Proceedings of the 27th Annual ACM Symposium on Applied Computing SAC 12, Page 1427-1434, <http://dl.acm.org/citation.cfm?id=2232005> , last accessed on Apr, 13, 2013.

- [Bar05] E. Barker, W. Barker, and A. Lee, NIST, “Guideline for Implementing Cryptography in the Federal Government”,

http://csrc.nist.gov/publications/nistpubs/800-21-1/sp800-21-1_Dec2005.pdf

- [Bli03] K. Blind, "Standards in the Service Sectors: An Explorative Study," Karlsruhe, 2003.

- [Bor11] N. Borenstein and J. Blake, "Cloud Computing Standards: Where's the Beef?" IEEE Internet Computing, vol. 15, no. 3, pp. 74-78, June 2011.

- [Bor12] J. Bort, “The 10 Most Important Companies In Cloud Computing”, <http://www.businessinsider.com/the-10-most-important-companies-in-Cloud-computing-2012-4?op=1>, - Last accessed on Apr, 21, 2013.

- [Bpe07] OASIS, WS-BPEL TC, <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf> - Last accessed on March 5, 2011.

- [Bra08] F. Braz, E.B. Fernandez, and M. VanHilst, "Eliciting security requirements through misuse activities" Procs. Of the 2nd Int. Workshop on Secure Systems Methodologies using Patterns (SPattern'07). In conjunction with the 4th International Conference on Trust, Privacy & Security in Digital Business(TrustBus'07), Turin, Italy, September 1-5, 2008. 328-333.
- [Bro09] J. Brodtkin, "10 Cloud computing companies to watch, Network World", <http://www.networkworld.com/supp/2009/ndc3/051809-Cloud-companies-to-watch.html>, - Last accessed on Apr, 21, 2013.
- [Bus06] F. Buschmann, R. Meunier, H. Rohner., P. Sommerlad, and M. Stal (1996). Pattern oriented software architecture: A system of patterns. West Sussex, England: Wiley.
- [But12] B. Butler, "*Gartner: Amazon, HP cloud SLAs are "practically useless"*", <http://www.networkworld.com/news/2012/120612-hp-amazon-cloud-264847.html>, Last access on June 28, 2013
- [Cdm13] Cloud Data Management Interface (CDMI), <http://www.snia.org/cdmi>, - Last accessed on Apr, 21, 2013.
- [Cho09] R. Chow et al., "Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control", ACM Workshop on Cloud Computing Security, Chicago, Illinois, November 2009,

<http://www2.parc.com/csl/members/eshi/docs/ccsw.pdf>
- [Cie11] ITU-T, X.1500 Cybersecurity information exchange techniques, <http://www.itu.int/ITU-T/recommendations/rec.aspx?id=11060> - Last accessed on Mar 05, 2013.
- [Cim12] DMTF, Cloud Infrastructure Management Interface 5 (CIMI) Model and RESTful HTTP-based Protocol –

- http://dmtf.org/sites/default/files/standards/documents/DSP0263_1.0.0.pdf
- Last accessed on Apr, 18, 2013.
- [Cio13] CIO, <http://www.cio.com/>
- [Cla10] Cloud Audit, CloudAudit 1.0,
<http://cloudataudit.googlecode.com/svn/trunk/docs/draft-off-cloudataudit.html> - Last accessed on Mar 05, 2013.
- [Clh13] Cloud Harmony, <http://cloudharmony.com/>
- [Cob11] M. Cobban, “What is BPEL and why is it so important to my business? “,
http://www.softcare.com/whitepapers/wp_what_is_bpel.php - Last accessed on March 9, 2011.
- [Coi12] Control Objectives for Information and Related Technology,
<http://www.isaca.org/COBIT/Pages/default.aspx> - Last accessed on Apr 22, 2013.
- [Csa11] Cloud Security Alliance, “Security Guidance for Critical Areas of Focus in Cloud Computing”, 2011
- [Csa13] Cloud Security Alliance, “The Notorious Nine: Cloud Computing Top Threats in 2013”. <https://Cloudsecurityalliance.org/research/top-threats/>
- [Csa13a] Cloud Security Alliance, <https://cloudsecurityalliance.org/>
- [Csa13b] Cloud Security Alliance, Cloud Controls Matrix (CCM), <https://cloudsecurityalliance.org/research/ccm/>
- [Csa13c] Cloud Security Alliance, CSA Security, Trust & Assurance Resources, <https://cloudsecurityalliance.org/star/>.

- [Csd13] Computer Security Division, NIST, Cryptographic Toolkit,
<http://csrc.nist.gov/groups/ST/toolkit/index.html>
- [Csw13] Cloud Standards Wiki,
http://Cloud-standards.org/wiki/index.php?title=Main_Page, -
 Last accessed on Apr, 21, 2013.
- [Ctp06] OASIS Standard, Web Services Security: X.509 Certificate Token Profile
 1.1, [https://www.oasis-open.org/committees/download.php/16785/wss-
 v1.1-spec-os-x509TokenProfile.pdf](https://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf) - Last accessed on Mar 05, 2013.
- [Cua13] Current Analysis, Enterprise Cloud Adoption,
<http://www.currentanalysis.com/custom/cloud/>, - Last accessed on Apr,
 22, 2013]
- [Cve11] ITU-T, X.1520 : Common vulnerabilities and exposures,
<http://www.itu.int/rec/T-REC-X.1520-201104-I/en> - Last accessed on Mar
 05, 2013. [100] NIST, Computer Security Incident Handling Guide,
<http://csrc.nist.gov/publications/drafts/800-61-rev2/draft-sp800-61rev2.pdf>
 - Last accessed on Mar 05, 2013.
- [Cvs12] ITU-T, X.1521 : Common vulnerability scoring system,
<http://www.itu.int/rec/T-REC-X.1521-201104-I/en> - Last accessed on Mar
 05, 2013.
- [Del05] N. Delessy, and E.B. Fernandez, "Patterns for the eXtensible Access
 Control Markup Language", in Proceedings of the 12th Pattern Languages
 of Programs Conference (PLoP2005), Monticello, Illinois, USA, 7-10
 September 2005.
[http://hillside.net/plop/2005/proceedings/PLoP2005_ndelessyandebfernan
 dez0_1.pdf](http://hillside.net/plop/2005/proceedings/PLoP2005_ndelessyandebfernan

 dez0_1.pdf)

- [Del07] N. Delessy, E.B. Fernandez, M.M. Larrondo-Petrie, "A pattern language for identity management", Procs. of the 2nd IEEE Int. Multiconference on Computing in the Global Information Technology (ICCGI 2007), March 4-9, Guadeloupe, French Caribbean.
- [Dia08] The DoD Information Assurance Certification and Accreditation Process (DIACAP), <http://www.diacap.org/>, - Last accessed on June, 08, 2013.
- [Dow11] S. Dowell, A. Barreto, J. Michael and M.T. Shing, "Cloud to Cloud Interoperability," in 6th International Conference on System of Systems Engineering (SoSE), Albuquerque, New Mexico, USA, 2011.
- [Dss09] NIST, FIPS 186-3: Digital Signature Standard (DSS), http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf - Last accessed on Mar 05, 2013.
- [Eau97] NIST, FIPS 196: Entity Authentication Using Public Key Cryptography, <http://csrc.nist.gov/publications/fips/fips196/fips196.pdf> - Last accessed on Mar 05, 2013.
- [Ees94] NIST, FIPS 185: Escrowed Encryption Standard (EES), <http://www.itl.nist.gov/fipspubs/fip185.htm> - Last accessed on Mar 05, 2013.
- [Emp12] EmpowerID, <http://www.empowerid.com/> - Last accessed on November 10, 2012
- [Fed13] The Federal Risk and Authorization Management Program (FedRAMP), http://www.gsa.gov/portal/category/102371?utm_source=OCSIT&utm_medium=print-radio&utm_term=fedramp&utm_campaign=shortcuts, - Last accessed on June, 08, 2013.

- [Fed13] The Payment Card Industry, Data Security Standards (PCI DSS Level 1), https://www.pcisecuritystandards.org/security_standards/, - Last accessed on June, 08, 2013.
- [Fer06a] E.B. Fernandez, M.M. Larrondo-Petrie, T. Sorgente, and M. VanHilst, "A methodology to develop secure systems using patterns", Chapter 5 in "Integrating security and software engineering: Advances and future vision", H. Mouratidis and P. Giorgini (Eds.), IDEA Press, 2006, 107-126.
- [Fer06b] E. B. Fernandez, M. VanHilst, M. M. Larrondo Petrie, S. Huang, "Defining Security Requirements through Misuse Actions", in Advanced Software Engineering: Expanding the Frontiers of Software Technology, S. F. Ochoa and G.-C. Roman (Eds.), International Federation for Information Processing, Springer, 2006, 123-137.
- [Fer06c] E.B. Fernandez, and N. Delessy. 2006. "Using patterns to understand and compare web services security products and standards", Proceedings of the Int. Conference on Web Applications and Services (ICIW'06), Guadeloupe, February 2006. IEEE Comp. Society, 2006.
- [Fer10] E.B. Fernandez, K. Hashizume, I. Buckley, M.M. Larrondo-Petrie, and M. VanHilst, "Web services security: Standards and products", Chapter 8 in "Web Services Security Development and Architecture: Theoretical and Practical Issues", Carlos A. Gutierrez, Eduardo Fernandez-Medina, and Mario Piattini (Eds.), IGI Global Group 2010. 152-177
- [Fer11] E.B.Fernandez and S. Mujica, "Model-based development of security requirements", accepted for the CLEI (Latin-American Center for Informatics Studies) Journal.
- [Fer12] E.B. Fernandez. Security Patterns in Practice: Building Secure Architectures Using Software Patterns; John Wiley & Sons: Hoboken, NJ, USA, 2012

- [Fer12a] E.B. Fernandez, O. Ajaj, I. Buckley, N. Delessy-Gassant, K. Hashizume, and M.M. Larrondo-Petrie, “A Survey of Patterns for Web Services Security and Reliability Standards”, *Future Internet* 2012, 4(2), 430-450. <http://www.mdpi.com/1999-5903/4/2/430>, - Last Access on August, 04, 2012

- [Fer13] E.B. Fernandez, *Security patterns in practice-Designing secure architectures using software patterns*, Wiley 2013 Series on Software Design Patterns (to appear).

- [Fis13] The Federal Information Security Management Act (FISMA), <http://csrc.nist.gov/groups/SMA/fisma/index.html>, - Last accessed on June, 08, 2013.

- [Gal91] NIST, FIPS 191: Guideline for the Analysis of Local Area Network Security, <http://www.itl.nist.gov/fipspubs/fip191.htm> - Last accessed on Mar 05, 2013.

- [Gar12] Gartner, Magic Quadrant for Cloud Infrastructure as a Service, http://www.gartner.com/DisplayDocument?ref=clientFriendlyUrl&id=2204015#document_history, Last access on June 28, 2013

- [Gol06] D. Gollmann, *Computer security* (2nd Ed.), Wiley, 2006.

- [Gua94] NIST, FIPS 190: Guideline for the Use of Advanced Authentication Technology Alternatives, <http://www.itl.nist.gov/fipspubs/fip190.htm> - Last accessed on Mar 05, 2013.

- [Gud04] M. Gudgin, “Using WS-Trust and WS-Secure Conversation”, MSDN 2004, <http://msdn.microsoft.com/en-us/library/ms996521.aspx> , Last accessed on August 16, 2012.

- [Has09] K. Hashizume, E.B. Fernandez. A Pattern for WS-Security. First IEEE Int. Workshop on Security Eng. Environments, Dec. 17-19, 2009, Shanghai, China.
- [Has09a] K. Hashizume, E.B.Fernandez, and S. Huang, "The WS-Security pattern", First IEEE Int. Workshop on Security Eng.Environments, Dec. 17-19, 2009Shanghai, China.
- [Has09b] K. Hashizume and E.B.Fernandez, "Symmetric Encryption and XML Encryption Patterns", Procs. of the 16th Conf. on Pattern Languages of Programs (PLoP 2009),

<http://portal.acm.org/citation.cfm?doid=1943226.1943243>
- [Has09c] K. Hashizume, E.B.Fernandez, and S. Huang, "Digital Signature with Hashing and XML Signature patterns", Procs. of the 14th European Conf. on Pattern Languages of Programs, EuroPLoP 2009.
- [Has13] K. Hashizume, G.R. David, E. Fernández-Medina, E.B. Fernandez, “An Analysis of Security issues for Cloud Computing”, accepted for the Journal of Internet Services and Applications, Springer. (SCOPUS)
- [Hic11] A.R. Hickey, “Researchers Uncover 'Massive Security Flaws' In Amazon Cloud”, <http://www.crn.com/news/cloud/231901911/researchers-uncover-massive-security-flaws-in-amazon-cloud.htm>, last accessed on Apr, 13, 2013
- [Hip96] The U.S. Health Insurance Portability and Accountability Act, <http://www.hhs.gov/ocr/privacy/> - Last accessed on Apr 22, 2013.
- [Hma08] NIST, FIPS 198-1: The Keyed-Hash Message Authentication Code (HMAC), http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf - Last accessed on Mar 05, 2013.

- [Hog11] M. Hogan, F. Liu, A. Sokol, and J. Tong, “NIST Cloud Computing Standards Roadmap.” National Institute of Standards and Technology, http://www.nist.gov/itl/cloud/upload/NIST_SP-500-291_Jul5A.pdf, Jul-2011
- [Hog11] M. Hogan, F. Liu, A. Sokol, and J. Tong, “NIST Cloud Computing Standards Roadmap.” National Institute of Standards and Technology, http://www.nist.gov/itl/cloud/upload/NIST_SP-500-291_Jul5A.pdf, Jul-2011
- [Hpc12] HPC12.
- [Ibm02] IBM, Security in a Web Services World: A Proposed Architecture and Roadmap, <http://download.boulder.ibm.com/ibmdl/pub/software/dw/library/ws-secmap.pdf> - Last accessed on March 30, 2012
- [Ibm04] IBM, “Web Services Security 2004”, <http://www.ibm.com/developerworks/library/specification/ws-secure/> – Last accessed at December, 15, 2009
- [Idc11] IDC, European Cloud Professional Services 2010 Market and 2011- 2015 Forecast, 2011.
- [Idc13] OASIS, Identity in the Cloud Technical Committee, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=id-Cloud, - Last accessed on Apr, 21, 2013.
- [Ide12] H.A.W Ideler, “Cryptography as a service in Cloud Computing environment”, Master thesis, Eindhoven University of Technology, 2012, <http://alexandria.tue.nl/extra1/afstversl/wsk-i/ideler2012.pdf>
- [Iee13] Survey by IEEE and Cloud Security Alliance details importance and urgency of cloud computing security standards,

- <http://standards.ieee.org/news/2010/cloudcomp.html>, March 2010. - Last accessed on Mar 05, 2013.
- [Iet11] IETF, System for Cross-domain Identity Management (SCIM), <http://datatracker.ietf.org/wg/scim/charter/>.
- [Inn07] InnoQ, “Web Services Standards Overview”, Version 3.0”, <http://www.innoq.com/soa/ws-standards/poster/innoQ%20WS-Standards%20Poster%202007-02.pdf>, - Last accessed on March 30, 2013.
- [Ipk08] IETF, RFC5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, <http://www.rfc-editor.org/rfc/rfc5280.txt> - Last accessed on Mar 05, 2013.
- [Iso05] International Organization of Standardization (ISO27001), <http://www.27000.org/iso-27001.htm>, - Last accessed on Apr 22, 2013.
- [Iws12] InformationWeek Standardization Survey
- [Jan11] W. Jansen, and T. Grance, NIST, “Guidelines on Security and Privacy in Public Cloud Computing”, <http://csrc.nist.gov/publications/nistpubs/800-144/SP800-144.pdf>- Last accessed on April 15, 2013
- [Joh06] S. Johnston, “Modeling security concerns in service-oriented architectures”, <http://www.ibm.com/developerworks/rational/library/4860.html>, last retrieved July 04, 2011.
- [Jts10] Joint Task Force Transformation Initiative, “Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach”, NIST Special Publication 800-37, Revision 1, <http://csrc.nist.gov/publications/nistpubs/800-37-rev1/sp800-37-rev1-final.pdf>

- [Kmi10] OASIS, Key Management Interoperability Protocol (KMIP) TC, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=kmip - Last accessed on Mar 05, 2013.
- [Kos12] A.W. Kosner, “Amazon Cloud Goes Down Friday Night, Taking Netflix, Instagram and Pinterest with It”, Forbes, <http://www.forbes.com/sites/anthonykosner/2012/06/30/amazon-cloud-goes-down-friday-night-taking-netflix-instagram-and-pinterest-with-it/>
- [Kpm10] KPMG, “From Hype to Future: KPMG’s 2010 Cloud Computing Survey.” 2010. Available: <http://www.techrepublic.com/whitepapers/from-hype-to-future-kpmgs-2010-Cloud-Computing-survey/2384291> - Last accessed on March 30, 2013.
- [Ktp07] OASIS Standard, Web Services Security: Kerberos Token Profile 1.1, <https://www.oasis-open.org/committees/download.php/16788/wss-v1.1-spec-os-KerberosTokenProfile.pdf> - Last accessed on Mar 05, 2013.
- [Kum12] A. Kumar and E.B. Fernandez, “A Security Pattern for the Transport Layer Security (TLS) Protocol”, 19th. Int. Conference on Pattern Languages of Programs (PLoP2012).
- [Kup11] M. Kuppinger, M., “SCIM – will SPML shortcomings be reinvented?” <http://blogs.kuppingercole.com/kuppinger/2011/04/23/scim-will-spml-shortcomings-be-reinvented/>.
- [Lin09] D.S. Linthicum, Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-Step Guide, Pearson Education (2009).
- [Liu05] H. Liu H, S. Pallickara, and G. Fox, ” Performance of Web Services Security” Proceedings of the 13th Annual 13th Mardi Gras Conference, Baton Rouge, Louisiana, February 3-5, 2005. <http://grids.ucs.indiana.edu/ptliupages/publications/WSSPerf.pdf>.

- [Liw09] W. Li and L. Ping, "Trust Model to Enhance Security and Interoperability of Cloud Environment," in Proceedings of the 1st International Conference on Cloud Computing, Beijing, China, 2009, pp. 69–79.
- [Mac09] G. S. Machado, D. Hausheer and B. Stiller, "Considerations on the Interoperability of and between Cloud Computing Standards," in 27th Open Grid Forum (OGF27), G2C-Net Workshop: From Grid to Cloud Networks, Banff, Canada, 2009.
- [Mad03] P. Madsen, "WS-Trust: Interoperable Security for Web Services", <http://www.xml.com/pub/a/ws/2003/06/24/ws-trust.html>- Last accessed on November 30, 2012
- [Mic06] Microsoft Corporation, .NET Framework Class Library,
<http://msdn.microsoft.com/en-us/library/ms951274.aspx> – Last accessed at March, 13, 2010.
- [Mic12] Active Directory Federation Services (ADFS),
<http://msdn.microsoft.com/en-us/library/bb897402.aspx> - Last accessed on November 10, 2012
- [Msr06] NIST, FIPS 200: Minimum Security Requirements for Federal Information and Information Systems,
<http://csrc.nist.gov/publications/fips/fips200/FIPS-200-final-march.pdf> - Last accessed on Mar 05, 2013.
- [Nis13] NIST, Final Version of NIST Cloud Computing Definition Published, <http://www.nist.gov/itl/csd/Cloud-102511.cfm>, - Last accessed on Apr, 21, 2013.
- [Nis13a] NIST, Inventory of Standards Relevant to Cloud Computing, <http://collaborate.nist.gov/twiki-cloud->

- computing/bin/view/CloudComputing/StandardsInventory - Last accessed on Apr 21, 2013.
- [Nis13a] NIST, Inventory of Standards Relevant to Cloud Computing, <http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/StandardsInventory> - Last accessed on Apr 21, 2013.
- [Nis13b] NIST, NIST Cloud Computing Collaboration Site, <http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/WebHome> - Last accessed on Mar 05, 2013.
- [Oap10] OAuth, OAuth (Open Authorization Protocol), <http://tools.ietf.org/html/rfc5849> - Last accessed on Mar 05, 2013.
- [Obr07] L. O'Brien, L. Bass, and P. Merson, "Quality Attributes and Service Oriented Architectures", Proceedings of International Workshop on Systems Development in SOA Environments (SDSOA'07), <http://dl.acm.org/citation.cfm?id=1270235.1270300&coll=DL&dl=ACM&CFID=326465762&CFTOKEN=50710868>
- [Occ11] Open Cloud Computing Interface (OCCI), <http://occi-wg.org/> - Last accessed on Apr, 18, 2013.
- [Occ13] The Open Cloud Computing Interface, <http://occi-wg.org/>, - Last accessed on Apr, 21, 2013.
- [Ogf07] Open Grid Forum, Usage Record – Format Recommendation , <http://www.ogf.org/documents/GFD.98.pdf> - Last accessed on Apr, 18, 2013.
- [Oid07] OAuth, OpenID Authentication, http://openid.net/specs/openid-authentication-2_0.html - Last accessed on Mar 05, 2013.

- [Omg13] Object Management Group: Business Process Management Initiative, <http://www.bpmn.org/>
- [Opg13] The Open Group, The Open Group Publishes New Standards for SOA and Cloud, <http://www3.opengroup.org/news/press/open-group-publishes-new-standards-soa-and-Cloud> , - Last accessed on Apr, 21, 2013.
- [Osf13] The Open Stack Foundation, <http://www.openstack.org/>, - Last accessed on Apr, 21, 2013.
- [Osp13] Amazon Web Services: Overview of Security Process, http://awsmedia.s3.amazonaws.com/pdf/AWS_Security_Whitepaper.pdf - Last accessed on April, 08, 2013.
- [Ost01] D. Osterwalder, “Trust Through Evaluation and Certification?”, Social Science Computer Review, 19, no. 1, Sage Publications, Inc., Spring 2001, pp. 32-46.
- [Ovf13] DMTF, Open Virtualization Format, http://dmtf.org/sites/default/files/standards/documents/DSP0243_2.0.0.pdf , - Last accessed on Apr, 18, 2013.
- [Ovf13] DMTF, Open Virtualization Format, <http://www.dmtf.org/standards/ovf>, - Last accessed on Apr, 21, 2013.
- [Pci10] PCI, PCI Data Security Standard, https://www.pcisecuritystandards.org/documents/pci_dss_v2.pdf - Last accessed on Mar 05, 2013.
- [Pci13] The Payment Card Industry, Data Security Standards (PCI DSS Level 1), https://www.pcisecuritystandards.org/security_standards/, - Last accessed on June, 08, 2013.

- [Per12] J.C. Perez, "Google Fixes Gmail Outage That Affected Millions of Users".
http://www.computerworld.com/s/article/9226281/Google_fixes_Gmail_outage_that_affected_millions, 17 April 2012.
- [Piv06] NIST, FIPS 201: Personal Identity Verification (PIV) of Federal Employees and Contractors, <http://csrc.nist.gov/publications/fips/fips201-1/FIPS-201-1-chng1.pdf> - Last accessed on Mar 05, 2013.
- [Pki04] IETF, X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile, <http://csrc.nist.gov/publications/fips/fips201-1/FIPS-201-1-chng1.pdf> - Last accessed on Mar 05, 2013.
- [Pst13] OASIS, OASIS Provisioning Services TC, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=provision.
- [Pwg13] IEEE, IEEE P2301 Working Group (Cloud Profiles), <http://grouper.ieee.org/groups/2301/>, - Last accessed on Apr, 21, 2013.
- [Rad12] RadiantOne Cloud Federation Service,
<http://www.radiantlogic.com/products/radiantone-cfs/>
- [Ras13] Amazon Web Services: Risk and Compliance, http://media.amazonwebservices.com/AWS_Risk_and_Compliance_Whitepaper.pdf - Last accessed on April, 08, 2013.
- [Rel07] OASIS Standard, Web Services Security Rights Expression Language (REL) Token Profile 1.1, <https://www.oasis-open.org/committees/download.php/16687/oasis-wss-rel-token-profile-1.1.pdf> - Last accessed on Mar 05, 2013.
- [Res00] The Representational State Transfer (REST),
http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm - Last accessed on Apr, 18, 2013.

- [Rid12] ITU-T, X.idmcc – Requirement of IdM in Cloud Computing, http://www.itu.int/itu-t/workprog/wp_item.aspx?isn=7865 - Last accessed on Mar 05, 2013.
- [Ros12] D. G. Rosado, R. Gómez, D. Mellado, and E. Fernández-Medina, “Security Analysis in the Migration to Cloud Environments,” *Future Internet*, vol. 4, no. 2, pp. 469–487, May 2012.
- [Saf13] OASIS, Symptoms Automation Framework (SAF), https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=saf, - Last accessed on Apr, 21, 2013.
- [Sam05] OASIS, Security Assertion Markup Language (SAML) SAML2.0, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=kmip - Last accessed on Mar 05, 2013.
- [Sam06] OASIS Standard, Web Services Security: SAML Token Profile 1.1, <https://www.oasis-open.org/committees/download.php/16768/wss-v1.1-spec-os-SAMLSecurityTokenProfile.pdf> - Last accessed on Mar 05, 2013.
- [Sap13] SAP AG, WS-SecureConversation, http://help.sap.com/saphelp_nwpi711/helpdata/en/48/aea404ac5a3206e1000000a42189c/content.htm.
- [Sas92] Statement on Auditing Standards (SAS no.700, http://sas70.com/sas70_overview.html - Last accessed on Apr, 18, 2013.
- [Sbp13] Amazon Web Services: AWS Security Best Practices, http://awsmedia.s3.amazonaws.com/pdf/AWS_Security_Whitepaper.pdf - Last accessed on April, 08, 2013.
- [Sca12] NIST, Security Content Automation Protocol (SCAP), <http://scap.nist.gov/> - Last accessed on Mar 05, 2013.

- [Sch06] M. Schumacher, E.B. Fernandez, D. Hybertson, F. Buschmann, and P. Sommerlad, Security Patterns: Integrating security and systems engineering, Wiley 2006.
- [Shi05] X.Shi, W. Han, Y. Li, and Y. Huang, “Integrated business-process-driven design for service-oriented enterprise applications”, J. Pervasive Computing & Comm., Vol 1, No 1, March 2005.
- [Shs12] NIST, FIPS 180-4: Secure Hash Standard (SHS),

<http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf> - Last accessed on Mar 05, 2013.
- [Sii13] IEEE, P2302 - Standard for InterCloud Interoperability and Federation (SIIF), <http://standards.ieee.org/develop/project/2302.html> - Last accessed on Apr, 21, 2013.
- [Sni10] SNIA, Cloud Data Management Interface (CDMI),

http://www.snia.org/tech_activities/standards/curr_standards/cdmi, - Last accessed on Apr, 18, 2013.
- [Soa07] The World Wide Web Consortium, SOAP Version 1.2, <http://www.w3.org/TR/soap/>, - Last accessed on Apr, 18, 2013.
- [Soc13a] The SSAE16 Auditing Standard, <http://www.ssaе-16.com/>, - Last accessed on June, 08, 2013.
- [Soc13b] SOC 2 Report, <http://www.ssaе-16.com/soc-2/>, - Last accessed on June, 08, 2013.
- [Sos10] D. Sosnoski, Java Web Services: WS-Trust and WS-Secure Conversation, IBM May 2010.<http://www.ibm.com/developerworks/java/library/j-jws15/index.html>. Last accessed August 17, 2012.

- [Sox02] The Sarbanes-Oxley Act of 2002, <http://www.soxlaw.com/> - Last accessed on Apr, 18, 2013.
- [Spm12] OASIS, Service Provisioning Markup Language (SPML), <https://www.oasis-open.org/standards#spmlv2.0> - Last accessed on Mar 05, 2013.
- [Src01] NIST, FIPS 140-2: Security Requirements for Cryptographic Modules, <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf> - Last accessed on Mar 05, 2013.
- [Src01] NIST, FIPS 140-2: Security Requirements for Cryptographic Modules, <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf> - Last accessed on Mar 05, 2013.
- [Ssc04] NIST, FIPS 199: Standards for Security Categorization of Federal Information and Information Systems,

<http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf> - Last accessed on Mar 05, 2013.
- [Ssl94] NIST, FIPS 188: Standard Security Label for Information Transfer, <http://www.itl.nist.gov/fipspubs/fip188.htm> - Last accessed on Mar 05, 2013.
- [Ssr13] Secure Systems Research Group (SSRG), Florida Atlantic University, <http://www.security.ceecs.fau.edu/> - last accessed at July, 01, 2013.
- [Sum12] Sumastre, G.M., "The Top Cloud Computing Mega-Vendors", <http://www.trainsignal.com/blog/top-Cloud-computing-vendors>, - Last accessed on Apr, 21, 2013.
- [Sym12] Symantec,

https://www4.symantec.com/mktginfo/whitepaper/TheSecureCloudBestPracticesforCloudAdoption_cta52644.pdf

- [Tak10] H. Takabi, J. Joshi and G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," *Security & Privacy, IEEE*, vol. 8, no. 6, pp. 24-31, Nov.-Dec. 2010.
- [Tec13] TechTarget, <http://www.techtarget.com/>
- [Tfd13] The Free Dictionary, <http://www.thefreedictionary.com/de+facto>. - Last accessed on Mar 05, 2013.
- [Tiv12] IBM Tivoli Federated Identity Manager, <http://www-01.ibm.com/software/tivoli/products/federated-identity-mgr/> - Last accessed on November 10, 2012
- [Tls08] IETF, The Transport Layer Security (TLS) Protocol Version 1.2, <http://tools.ietf.org/html/rfc5246> - Last accessed on Mar 05, 2013.
- [Tos12] OASIS, Topology and Orchestration Specification for Cloud Applications Version 1.0,

<http://docs.oasis-open.org/tosca/TOSCA/v1.0/csprd01/TOSCA-v1.0-csprd01.pdf> - Last accessed on Mar 05, 2013.
- [Uci12] OASIS, Use Cases for Identity Management in the Cloud, <https://wiki.oasis-open.org/id-cloud/> - Last accessed on Mar 05, 2013.
- [Utp06] OASIS Standard, Web Services Security: Username Token Profile 1.1, <https://www.oasis-open.org/committees/download.php/16782/wss-v1.1-spec-os-UsernameTokenProfile.pdf> - Last accessed on Mar 05, 2013.
- [Van09] M. VanHilst, E.B.Fernandez, and F. Braz, "A multidimensional classification for users of security patterns", *Journal of Research and Practice in Information Technology*, vol. 41, No 2, May 2009, 87-97. (ISI)

- [Whi12] Z. Whittaker, Amazon cloud down; Reddit, Github, other major sites affected, <http://www.zdnet.com/amazon-cloud-down-reddit-github-other-major-sites-affected-7000006166/>
- [Wik13] Wikipedia, Web API, http://en.wikipedia.org/wiki/Web_API
- [Wik13] Wikipedia, Wikimedia Foundation, Inc. Business Process Execution Language, http://en.wikipedia.org/wiki/Business_Process_Execution_Language
- [Wme04] W3C, “Web Services Metadata Exchange”, <http://www.w3.org/TR/ws-gloss/> – Last accessed at March, 15, 2013
- [Wme09] W3C, Web Services Metadata Exchange (WS-MetadataExchange), W3C Working Draft 17 March 2009, <http://www.w3.org/TR/2009/WD-ws-metadata-exchange-20090317/> - Last Accessed on September, 29, 2012
- [Wsb07] OASIS, WS-BPEL TC, <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf> - Last accessed on March 5, 2011
- [Wsc09] OASIS Standard, WS-SecureConversation 1.4. <http://docs.oasis-open.org/ws-sx/ws-secureconversation/v1.4/os/ws-secureconversation-1.4-spec-os.pdf> - Last Accessed on February, 29, 2012
- [Wsf09] OASIS, “Web Services Federation Language (WS-Federation) Version 1.2”, <http://docs.oasis-open.org/wsfed/federation/v1.2/ws-federation.pdf> - Last accessed on March 10, 2013.
- [Wsp07] The World Wide Web Consortium, Web Services Policy 1.5 – Framework, 4 September 2007, <http://www.w3.org/TR/ws-policy/> - Last accessed on Mar 05, 2013.
- [Wsp07] W3C, “Web Services Policy 1.5 Framework” (2007), <http://www.w3.org/TR/ws-policy/>– Last accessed at March, 15, 2013

- [Wsp09] OASIS Standard, WS-SecurityPolicy 1.3. 25 April 2012, OASIS Standard incorporating Approved Errata, <http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.3/errata01/os/ws-securitypolicy-1.3-errata01-os-complete.html> - Last accessed on Mar 05, 2013.
- [Wss04] OASIS Standard, Web Services Security: (WS-Security 2004), <https://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf> - Last accessed on Mar 05, 2013.
- [Wss06] OASIS, “Web Services Security: (WS-Security 2004)”, <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf> - Last accessed on March 07, 2013.
- [Wss07] OASIS Standard, WS-SecureConversation 1.3, <http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.pdf> - Last accessed on Mar 05, 2013.
- [Wst07] OASIS Standard, WS-Trust 1.4, <http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/os/ws-trust-1.4-spec-os.pdf> - Last accessed on March 07, 2012
- [Wst09] OASIS, “WS-Trust 1.4”, <http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/os/ws-trust-1.4-spec-os.pdf> - Last accessed on March 07, 2013.
- [Xac05] OASIS Standard, eXtensible Access Control Markup Language (XACML) Version 3.0, 22 January 2013, <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.pdf> - Last accessed on Mar 05, 2013.
- [Xds08] W3C, XML Signature Syntax and Processing (Second Edition), 10 June 2008, <http://www.w3.org/TR/xmlsig-core> - Last Accessed on February, 29, 2012.
- [Xes02] The World Wide Web Consortium, XML Encryption Syntax and Processing, <http://www.w3.org/TR/xmlenc-core/> - Last accessed on Mar 05, 2013.

- [Xml06] The World Wide Web Consortium, Extensible Markup Language (XML) 1.1 (Second Edition), <http://www.w3.org/TR/xml11/> - Last accessed on Apr, 18, 2013.
- [Xpl10] The World Wide Web Consortium, XML Path Language (XPath) 2.0 (Second Edition), <http://www.w3.org/TR/xpath20/> - Last accessed on Apr, 18, 2013.
- [Xss08] W3C, XML Signature Syntax and Processing (Second Edition), 10 June 2008, <http://www.w3.org/TR/xmldsig-core/> - Last Accessed on September, 29, 2012

Filename: OAjaj-DissertationJuly17-13.doc
Directory: C:\Users\ojaj\Documents
Template: C:\Users\ojaj\AppData\Roaming\Microsoft\Templates\Normal.do
tm
Title:
Subject:
Author: Ola Ajaj
Keywords:
Comments:
Creation Date: 11/17/2012 11:16:00 AM
Change Number: 212
Last Saved On: 8/15/2013 2:59:00 PM
Last Saved By: Ola Ajaj
Total Editing Time: 7,799 Minutes
Last Printed On: 8/15/2013 3:02:00 PM
As of Last Complete Printing
Number of Pages: 232
Number of Words: 44,596 (approx.)
Number of Characters: 254,198 (approx.)