TOWARDS A SECURITY FRAMEWORK FOR THE SEMANTIC WEB

by

IBRAHIM RAJAB MBAYA

submitted in fulfilment of the requirements
for the degree of

MASTER OF SCIENCE

in the subject

COMPUTER SCIENCE

at the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR: Prof. ALTA J. VAN DER MERWE
JOINT SUPERVISOR: Dr. AURONA J. GERBER

NOVEMBER 2007

ACKNOWLEDGEMENTS

I would like to thank the almighty God, the creator of the universe, for making this project possible under difficult and challenging conditions.

I would also like to thank my supervisors Prof. Alta Van der Merwe and Dr Aurona Gerber, for their guidance, support, unbiased criticism and patience in the whole process of this research.

Special thanks go to my parents and siblings, for their encouragement and advice that kept me going at the hardest time of my life. Your financial, moral, and emotional support made this research possible.

Lastly but not least, I would like to thank the Department of Correctional Services for allowing me to pursue this research.

ABSTRACT

With the increasing use of the Web and the need to automate, interoperate, and reason about resources and services on the Web, the Semantic Web aims to provide solutions for the future needs of World Wide Web computing. However, the autonomous, dynamic, open, distributed and heterogeneous nature of the Semantic Web introduces new security challenges. Various security standards and mechanisms exist that address different security aspects of the current Web and Internet, but these have not been integrated to address security aspects of the Semantic Web specifically. Hence, there is a need to have a security framework that integrates these disparate security tools to provide a holistic, secure environment for the Semantic Web.

This study proposes a security framework that provides various security functionalities to Semantic Web entities, namely, agents, Web services and Web resources. The study commences with a literature survey carried out in order to establish security aspects related to the Semantic Web. In addition, requirements for a security framework for the Semantic Web are extracted from the literature. This is followed by a model-building study that is used to compile a security framework for the Semantic Web. In order to prove the feasibility thereof, the framework is then applied to different application scenarios as a proof-of-concept. Following the results of the evaluation, it is possible to argue that the proposed security framework allows for the description of security concepts and service workflows, reasoning about security concepts and policies, as well as the specification of security policies, security services and security mechanisms. The security framework is therefore useful in addressing the identified security requirements of the Semantic Web.

Keywords: Semantic Web; security; agents; Web services; security framework

ABBREVIATIONS

Abbreviation	Description
AAA	Authentication Authorisation Accounting
ACL	Access Control List
ACM	Association for Computing Machinery
API	Application Program Interface
BPEL	Business Process Execution Language
BPEL4WS	Business Process Execution Language for Web service
CBSE	Component Based Software Engineering
DAML	DARPA Agent Mark-up Language
DAML-S	DAML Services
HTML	Hypertext Mark-up Language
HTTP	Hypertext Transfer Protocol
IEEE	Institute for Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IPSec	Internet Protocol Security
ISO	International Standard Organisation
JAL	Juxtapose Abstract Layer
JXTA	Juxtapose
KAoS	Knowledgeable Agent-oriented System
KMC	Key Management Centre
LAN	Local Area Network
MAC	Message Authentication Code
MAMD	Multi-Agent Multi-Domain
OIL	Ontology Inference Layer
OWL	Ontology Web Language
OWL-DL	Ontology Web Language – Description Language
OWL-S	Ontology Web Language – Services
P3P	Platform for Privacy Preferences
PDA	Personal Data Assistant
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure

RDF	Resource Description Framework
RDF-S	RDF Schema
RIF	Rule Interchange Format
S/MIME	Secure Multipurpose Internet Mail Extensions
SAML	Security Assertion Mark-up Language
SOAP	Simple Object Access Protocol
SPKI	Simple Public Key Infrastructure
SSH	Secure Shell
SSL	Secure Socket Layer
SWRL	Semantic Web Rule Language
SWWS	Semantic Web-enabled Web Services
UDDI	Universal Description Discovery and Integration
UML	Unified Mark-up Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WS-BPEL	Business Process Execution Language for Web Service
WS-CDL	Web Service Choreography Description Language
WSCI	Web Service Choreography Interface
WSDL	Web Service Description Language
WSS	Web Service Security
WWW	World Wide Web
XACML	eXtensible Access Control Mark-up Language
X-KISS	XML Key Information Service Specification
XKMS	XML Key Management Specification
X-KRSS	XML Key Registration Service Specification
XML	eXtensible Mark-up Language
XMLDSIG	XML Digital Signature
XMLENC	XML Encryption
XML-NS	XML Namespace
XML-S	XML Schema

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1. BACKGROUND

The increased importance of data management in many organisations has resulted in the development of different technologies to achieve efficient access to data, sharing of data and extraction of information from data sources (Thuraisingham, 2003).

In the late 1980s and mid 1990s there were rapid increases in the deployment of the Internet and corporate databases (Finin & Joshi, 2002; Thuraisingham, 2003). Various tools have been developed to provide interoperability as well as warehousing between multiple data sources, and to extract information from databases and warehouses on the Web (Thuraisingham, 2003).

In the late 1990s, the inadequacy of the existing Web technologies to automate processing of Web resources was realised by Tim Berners-Lee (Thuraisingham, 2003). His conclusion was that one needs machine-understandable Web pages and the use of ontologies for information integration (Berners-Lee, 2000; Horrocks & Patel-Schneider, 2003; Thuraisingham, 2003). Hence, the concept of the Semantic Web, which would provide machine-understandable metadata about resources, was introduced (Berners-Lee, Hendler & Lassila, 2001; Fensel, 2002; Park, 2003).

The Semantic Web is the extension of the current World Wide Web (hereafter referred to as the Web) where resources are enriched with machine-understandable metadata that describe their meaning to enable easy processing of information by machines and software agents (Berners-Lee et al., 2001; Palmer, 2001; Park, 2003).

The Semantic Web is envisioned as enabling software agents and search engines to find and interpret Web contents much more quickly and accurately than is possible with current keyword-searching or data-mining techniques (Denker, Kagal, Finin, Paulocci & Sycara, 2003). The

Semantic Web is thus regarded as a technology that integrates information applications and systems to provide mechanisms for the realisation of Enterprise Information Systems (Gerber, Barnard & Van der Merwe, 2006).

Until this vision of the Semantic Web materialises, efforts that require finding data spread across the Web, or dynamically drawing inferences based on this data, will continue to be hindered by their reliance on task-specific frameworks (Finin & Joshi, 2002).

Recently there were significant advances in the development of infrastructure and technologies to support dynamic interaction, interoperation, discovery and reasoning about the Internet. Technologies such as RDF-S (Brickley & Guha, 2003), DAML-S (Ankolekar, Bursten, Hobbs, Lassila, Martin, Drew-McDermott, Narayanan, Paulocci, Payne, & Sycara, 2001), OWL (McGuinness & Van Harmelen, 2004), and SWRL (Horrocks, Patel-Schneider, Boley, Tabet, Grosof, & Dean, 2004) were introduced to provide dynamic and adaptive data management on the Web (Ashri, Payne, Marvin, Surridge, & Taylor, 2004).

The shift to a more autonomous system, where direct user intervention is not necessary when making decisions, creates new security challenges (Ashri et al., 2004). The main challenge is to provide security to the more autonomous systems that support complex and dynamic relationships between clients and service providers. Given the completely decentralised nature of the Web, the extremely large number of users, agents and services, and their heterogeneity, security becomes increasingly important, and it is crucial for the security mechanisms to be included in the development of the Semantic Web (Finin & Joshi, 2002).

The ability to handle security and privacy and to automate the security mechanisms is a key need for the success of the Semantic Web. The need for security increases with the recent increased interest in Web-based e-commerce, the amount of business that is transacted online and the explosion in the number of services available on the Web (Kagal et al., 2003).

The use of conventional security technologies such as PKI, X.509, SSL, etc. to provide security on the Semantic Web is insufficient owing to the dynamic and adaptive nature of the Semantic Web (Ashri et al., 2004). Moreover, these technologies are XML-based, which makes them suitable for authentication and accountability rather than authorisation as desired for the Semantic Web (Kagal et al., 2003). It is necessary to describe and reason about security requirements at the semantic level in order to provide dynamic and adaptive security mechanisms to the Semantic Web (Ashri et al., 2004).

Various attempts have been made to develop security infrastructures and mechanisms for the Semantic Web, including the work by Farkas and Huhns (2002), which suggests that security models and tools for the underlying technologies such as XML, RDF, and OWL need to be developed to provide security for the Semantic Web. They argue that security requirements are driven by functionality, collaboration and organisational needs. Furthermore, Farkas and Huhns (2002) argue that there is a need for flexible models for Web access control that support fine-grained data granularity, that accommodate a wide range of policies, that are suitable for dynamic, decentralised and open environments, that are scalable, and that preserve the semantic consistency of data and limit illegal inferences.

The study in security issues for the Semantic Web by Thuraisingham (2003) suggests that security cuts across all layers of the Semantic Web. Therefore, to achieve a secure Semantic Web, one needs to secure components of the Semantic Web, secure integration of components, secure information integration and examine trust issues in relation to the Semantic Web.

The work of Denker et al. (2003) addresses the issue of annotating service descriptions with information relating to their security requirements and capabilities. This information can be used to ensure that clients and service providers meet one another’s security requirements. They argue that such annotation is useful in enabling reasoning about security at the semantic level.

Kagal et al. (2003) suggest the use of semantic policy language for defining security requirements, and distributed policy management as an alternative to authentication and access control schemes for the Semantic Web. They propose a policy engine that interprets and reasons about identified policies (security, privacy, management, and conversation). This work follows a more decentralised and adaptive model where the use of speech acts supports the dynamic modification of policies.

Ashri et al. (2004) propose the use of conventional security solutions with the ability to reason about security at the semantic level to achieve a secure Semantic Web. They further suggest the use of security policies (Kagal et al., 2003; Bhargavan, Fournet & Gordon, 2004) to describe security requirements and capabilities of entities. They also suggest the use of Semantic Firewall (Uszok, Bradshaw, & Jeffers, 2004a) to enforce the security policies. The security infrastructure suggested should have description capabilities, reasoning capabilities and infrastructure capabilities.

Other programs aimed at Semantic Web security adapt data exchange formats and protocols related to security in distributed systems to the Semantic Web. Such programs include:

- XMLEnc (Imamura, Dillaway & Simon, 2002; Klyne, 2002), which supports end-to-end encryption of XML objects
- XMLDSig (Bartel, Boyer, Fox, Lamacchia, & Simon, 2002; Klyne, 2002), a mechanism to sign and verify an entity unambiguously

- XKMS (Ford., Hallam-Baker, Fox, Dillaway, Lamacchia, Epstein, & Lapp, 2001), a mechanism for key distribution and verification
- XACML (Klyne, 2002), a language in XML for expressing access policies. XACML allows control over actions and supports resolution of conflicts.
- P3P (Klyne, 2002), which enables websites to describe their privacy policies and allows browsers to reason about these policies to decide whether they match users' preferences
- SAML (Klyne, 2002; Cover, 2006), a framework for exchanging security information such as authentication and authorisation decisions.

All these programs address different security issues and make use of different technologies. However, each program makes an important contribution towards dealing with security aspects of the Semantic Web by making use of single Semantic Web technologies only. Owing to the openness of the Semantic Web, a solution for the Semantic Web is not expected to be adopted from a single security standard alone (Denker et al., 2003).

From the literature surveyed, it is possible to argue that no formal integrative framework for different security-related approaches exists. This study therefore aims at compiling an integrative security framework for the Semantic Web that will take advantage of the current Semantic Web technologies and security standards that are implementable using the existing Semantic Web technologies.

1.2. RESEARCH PROBLEM

The need to have an integrative security framework for the Semantic Web that addresses the security aspects of the Semantic Web forms the basis of the research problem.

The research questions will be formulated in such a way that the questions will guide the research process to reach the research objectives. Chapters 4, 5 and 6 of the dissertation will provide specific answers that contribute to the overall answer to the main research question.

The main research question for this study is: **How can a security framework for the Semantic Web be constructed?**

In order to answer this question, one needs to answer the following sub-questions:

Sub-question 1: **What security aspects are related to the Semantic Web?** The aim of this question is to identify security aspects related to the Semantic Web and to provide a theoretical context for the research.

Sub-question 2: **What are the requirements of a security framework for the Semantic Web?** This question seeks to establish characteristics that will be used as requirements for a security framework for the Semantic Web.

Sub-question 3: **What are the components that we can use from existing security frameworks?** This question is intended to evaluate existing security frameworks and to determine components applicable to the Semantic Web.

Sub-question 4: **What are the components of a security framework for the Semantic Web?** The aim of this question is to identify elements or components that are needed to compile a security framework for the Semantic Web.

1.3. RESEARCH OBJECTIVES

The objectives of this study are formulated in such a way that they provide the means for answering the proposed research questions. The objectives of this study are:

- to describe security aspects related to the Semantic Web
- to establish the requirements of a security framework for the Semantic Web
- to determine components from existing security frameworks that can be used for the Semantic Web
- to establish components of a security framework for the Semantic Web
- to demonstrate the applicability of a security framework for the Semantic Web.

1.4. RESEARCH METHODOLOGY

Based on the research problem, a security framework for the Semantic Web will be compiled. In order to be able to compile the security framework for the Semantic Web, a qualitative research approach will be followed.

The research approach will consist of a literature review, where security aspects related to the Semantic Web and existing security frameworks will be thoroughly studied. The literature review will also include document analysis for the extraction of supporting information such as the requirements of a security framework for the Semantic Web.

The literature review will be followed by arguments to establish components of a security framework for the Semantic Web. Arguments will also be used to ascertain whether a differentiated framework is needed.

A model-building study will follow, in which a security framework for the Semantic Web will be developed by performing analogical reasoning from the existing security frameworks. The framework will also be based on the arguments constructed and propositions made from the literature.

Lastly, the framework will be applied to different application scenarios as a **proof-of-concept**.

1.5. DELIMITATION AND SCOPE OF THE STUDY

1.5.1. Research assumptions

The Semantic Web is a vision in which *machines* will be able to access and process information rather than *humans* (Berners-Lee et al., 2001), in contrast to the current Web, where humans are able to access and process information. The research therefore makes an assumption that the term Semantic Web means a hybrid Web by which both humans and machines can access and process information.

The World Wide Web Consortium (W3C)[1] plays an important role in providing specifications for Semantic Web technologies and management of such standards. The study therefore assumes that Semantic Web technologies are only those standards recommended or adopted by the W3C (W3C, 2004d).

1.5.2. Scope

The study will cover security aspects implementable by the current Semantic Web technologies as outlined in Chapter 2. Various security frameworks will be discussed and evaluated against the requirements of a security framework for the Semantic Web. A security framework for the Semantic Web will be proposed and its application to different use case scenarios will be implemented as a proof-of-concept.

[1] http://www.w3c.org/

The following issues do not fall within the scope of this research.

- The study will not suggest new technologies such as language, ontology, etc. for the Semantic Web.
- The study will not discuss architectural issues of the Semantic Web that might affect security aspects of the Semantic Web, for example, layering issues (Patel-Schneider & Fensel, 2002).
- The study will not suggest new and specific countermeasures, but will rather propose a complete security framework where different countermeasures will reside.

Therefore, the study uses the *established technologies* (Gerber et al., 2006) and existing countermeasures for its implementation, instead of establishing new technologies or proving the feasibility of a new technology or countermeasure.

1.5.3. Limitations

The following methodological limitations apply to the study.

- No fully-fledged analysis of the implementation issues of the components required for the proposed security framework will be carried out.
- The study will not involve the evaluation of the application scenarios in terms of suitable implementation infrastructure, performance issues, etc.

1.6. WORKING DEFINITIONS

The following key terms are defined in accordance with the context of this study.

- **Agents**: autonomous software entities that can interact with their environment, in this context the Semantic Web
- **Machines**: computers (including PDAs, cell phones) and computer programs that can perform tasks on the Web
- **Security frameworks**: *frameworks, models, architectures, infrastructures* and *approaches* that provide one or more security

services such as integrity, authentication, authorisation, confidentiality, and non-repudiation

- **Security**: the goal of attaining authenticity, confidentiality, integrity, non-repudiation, etc. in respect of computing assets such as data, Web resources and Web services
- **Semantic Web**: the extension of the World Wide Web with machine-readable information and automated services that assist in data interoperability across applications and organisations
- **Web resources**: objects that are accessible through the Web such as Web pages, documents, etc
- **Web services**: software applications that are accessible through the Web and that support direct interactions with other software agents.

1.7. OUTLINE OF CHAPTERS

The following table outlines the chapters of the dissertation and provides a brief description of the contents of each.

Table 1.1: Outline of chapters

CHAPTER	CHAPTER DESCRIPTION
1: INTRODUCTION	**Chapter 1: Introduction (this chapter)** Provides a background to the study, research problem, research objectives, research methodology, delimitation of the study, and the structure of the dissertation.
2: THEORETICAL FRAMEWORK	**Chapter 2: The Semantic Web and its security aspects** Provides an overview of the Semantic Web and a discussion of the security aspects related to the Semantic Web. Lastly, the discussion of existing security frameworks is given.
3: DESIGN AND METHODS	**Chapter 3: Research design and methodology** Provides a discussion of the research design followed to develop a security framework for the Semantic Web. A motivation for the selected research design is presented together with its limitations.

4: RESEARCH FINDINGS AND ANALYSIS	**Chapter 4: Analysis of existing security frameworks** The chapter extracts the requirements of a security framework for the Semantic Web, evaluates existing security frameworks, and establishes the essential components of a security framework.
	Chapter 5: A security framework for the Semantic Web The chapter presents a proposed security framework for the Semantic Web. Components and functionalities of the framework are explained.
	Chapter 6: Proof-of-concept scenarios The proposed security framework is applied to different application scenarios as a proof-of-concept.
5: CONCLUSION	**Chapter 7: Conclusion and contribution** The chapter discusses the contribution made by the study. It presents a summary of the study and its findings. It also provides recommendations for future research.
REFERENCES	List of literature referred to in the dissertation.

Figure 1.1 below (Dissertation map) illustrates the relationship between the research questions, research methods, research findings and chapters of the dissertation.

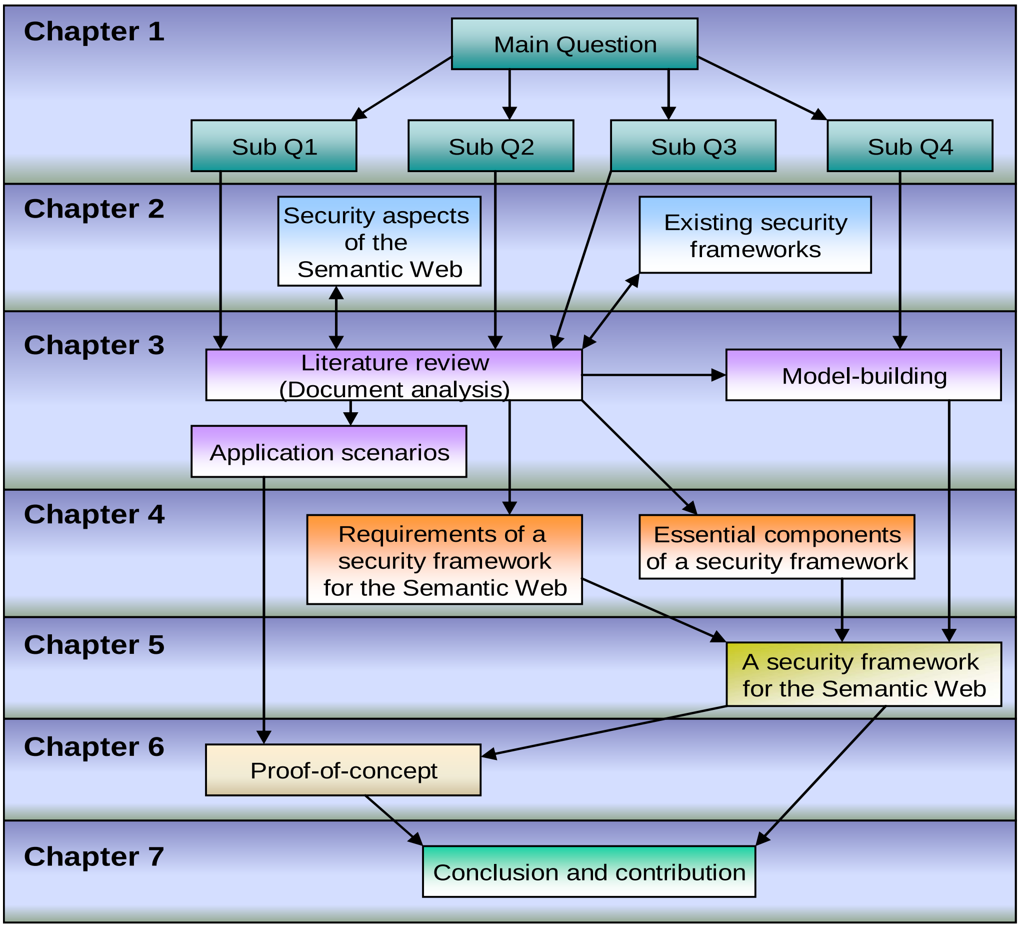

Chapter 1
Main Question
Sub Q1
Sub Q2
Sub Q3
Sub Q4
Chapter 2
Security aspects of the Semantic Web
Existing security frameworks
Chapter 3
Literature review (Document analysis)
Model-building
Application scenarios
Chapter 4
Requirements of a security framework for the Semantic Web
Essential components of a security framework
Chapter 5
A security framework for the Semantic Web
Chapter 6
Proof-of-concept
Chapter 7
Conclusion and contribution

Figure 1.1: Dissertation map

1.8. CONCLUSION

This chapter introduces this dissertation. It includes a detailed background study and discusses the rationale for the study. The research questions informing the study were formulated, followed by the research objectives. A research methodology for the execution of the research for the study is suggested before the presentation of the delimitations of the study. Lastly, the outline of the chapter is given with a dissertation map to facilitate the reader’s understanding of the structure of this document.

CHAPTER 2: THE SEMANTIC WEB AND ITS SECURITY ASPECTS

2.1. INTRODUCTION

In this chapter, the security aspects of the Semantic Web are established. In order to establish the security aspects of the Semantic Web, firstly an overview of the Semantic Web is given in Section 2.2 in order to provide a detailed understanding of the Semantic Web and its associated technologies and functionalities. A discussion of security aspects related to the Semantic Web follows in Section 2.3, where protected assets, various security threats and security services desired for the Semantic Web are discussed. Section 2.4 presents a discussion of existing security frameworks. Lastly, in Section 2.5 this chapter concludes with a summary of the security aspects for the Semantic Web.

2.2. THE SEMANTIC WEB

2.2.1. Background

Berners-Lee et al. (2001) presented a vision of a Web called the Semantic Web that is described as an information space usable by machines rather than humans. In the Semantic Web, a user would have personal software agents that would search Web resources and Web services, process information from multiple sources, exchange results with other software agents on behalf of users and present the results to the user, who would only have access to the results presented by his or her software agent.

Since the inception of the Semantic Web, various attempts have been made to develop technologies or languages required for realisation of the Semantic Web (Bray, Paoli, Sperberg-McQueen, Maler, & Yergeau, 2004; McGuinness & Van Harmelen, 2004; W3C, 2004a). With the guidance of the W3C (World Wide Web Consortium), the core technological building blocks for the Semantic Web are in place and available for developers (W3C, 2004d). In February 2004, the W3C announced that the Semantic Web had emerged as a commercial-grade infrastructure for sharing data on the Web (W3C, 2004d).

Although the Semantic Web concept captured the interest and imagination of a significant number of Web users, it is still primarily an international research effort whose goal is to make Web contents available for intelligent knowledge processing (Palmer, 2001; Euzenat & Napoli, 2003; Uschold, 2003; Grau, 2004). Currently the Semantic Web is an active discussion, research and development topic.

2.2.2. The Semantic Web definition, architecture and technologies

The Semantic Web is an extended Web of machine-readable information and automated services that extends beyond current capabilities of the World Wide Web (Berners-Lee et al., 2001). It is regarded as an information space usable by machines rather than humans. According to Berners-Lee (2006), the Semantic Web is a mechanism that assists data interoperability across applications and organisations. It is a set of interoperable standards for data, information and knowledge exchange, and for integration between applications and communities.

The Semantic Web enables software agents and search engines to find and interpret Web contents more quickly and with more accuracy than is possible with current keyword-searching or data-mining techniques (Denker et al., 2003). The main use of the Semantic web is to integrate diverse data sources intelligently into modern Information systems. The Semantic Web is used for integration and exchange of data, information, and knowledge across communities and applications.

The Semantic Web consists of various interoperable technologies that perform different functions within the context of the Semantic Web. In order to understand the purpose and functions of these technologies, one needs to investigate the architecture of the Semantic Web.

The Semantic Web architecture is generally presented as a layered architecture in which semantic language functionalities and technologies are layered into an increasingly expressive stack

(Berners-Lee et al., 2001; Berners-Lee, 2003; Berners-Lee, 2005; Berners-Lee, 2006). The purpose of the Semantic Web layered architecture is to depict the languages necessary for data interoperability between applications.

Different versions of the Semantic Web architecture were released by Tim Berners-Lee in order to organise the existing Semantic Web technologies and to identify functionalities of metadata languages used on the Semantic Web (Berners-Lee et al., 2001; Berners-Lee, 2003; Berners-Lee, 2005; Berners-Lee, 2006). Figures 2.1 to 2.4 depict the four versions of the Semantic Web architecture.

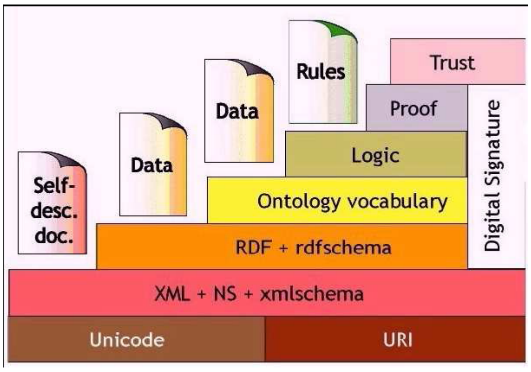


Figure 2.1: Semantic Web architecture (Berners-Lee et al., 2001)

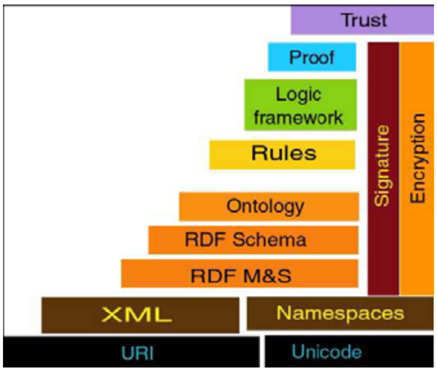


Figure 2.2: Semantic Web architecture (Berners-Lee, 2003)

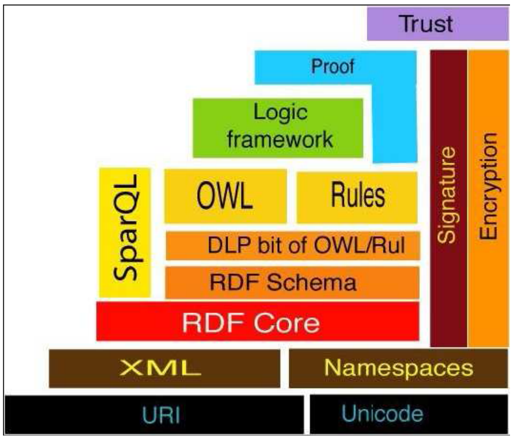


Figure 2.3: Semantic Web architecture (Berners-Lee, 2005)

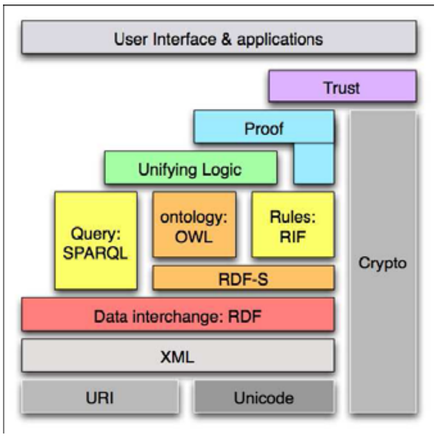


Figure 2.4: Semantic Web architecture (Berners-Lee, 2006)

From these four versions of the Semantic Web architecture, it is possible to observe that the layering issues of the Semantic Web have not yet stabilised. It is also evident that certain layers are termed according to their functionalities, for example, Trust, while others are termed according to the technologies used in that layer e.g. XML.

For the purpose of this study, the status model of the Semantic Web architecture proposed by Gerber et al. (2006) will be adopted. The status model is adopted because it clarifies the confusion in Semantic Web terminologies and in many questions arising from different versions of the Semantic Web architecture. Figure 2.5 below depicts the Semantic Web status model.

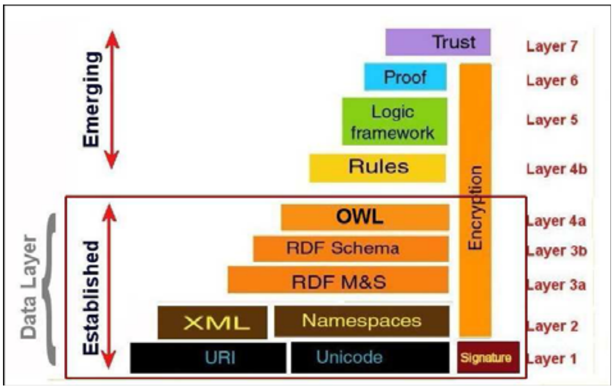


Figure 2.5: The Semantic Web architecture status model (Gerber et al., 2006)

The Semantic Web architecture is based on a hierarchy of *languages* (interoperable standards), each language both exploiting the features and extending the capabilities of the layer below it (Horrocks & Patel-Schneider, 2003; Horrocks, Parsia, Patel-Schneider, & Hendler, 2005). Each language either maintains or extends the syntax and semantics of the languages below it. In the context of this study the term languages is interchangeable with *standards* or *technologies* and it refers to tools that are required to enrich Web resources with machine-understandable metadata about their meaning.

This organisation of the languages required for metadata specification enables machines to process information on the Web efficiently (Berners-Lee et al., 2001). Languages that have been adopted by the W3C for the Semantic Web include Unicode, Uniform Resource

Identifier (URI), extensible mark-up language (XML), Resource Description Framework (RDF), RDF-Schema (RDF-S) and ontology Web language (OWL) (Berners-Lee, 2000; Berners-Lee et al., 2001; Horrocks et al., 2005).

The established technologies

The bottom four layers of the Semantic Web architecture are classified as the established technologies because the technologies used in these layers have been adopted or recommended by the W3C.

Unicode and *URI*

Unicode and Uniform Resource Identifier (URI) reside in layer 1 of the Semantic Web architecture (Berners-Lee et al., 2001; Berners-Lee, 2003). Layer 1 technologies of the Semantic Web provide a unique identification mechanism for the upper-layer language technologies (Gerber, 2007).

Unicode uniquely identifies the characters in all written languages by assigning a unique number to each character (UNICODE, 2006). It specifies a global character-encoding mechanism that allows data and text to be exchanged globally between different systems.

The Unicode consortium (UNICODE, 2004) manages the Unicode standard. The Unicode standard supports three encoding mechanisms: UTF-8, UTF-16, and UTF-32; hence, a data item can be encoded in a byte, word or double-word format (UNICODE, 2006). Unicode is supported in modern operating systems and browsers.

URI is used to uniquely identify an abstract or physical resource (Palmer, 2001; Berners-Lee, 2005). URI forms an ideal base technology upon which to build a global Web, as anything that has a URI is considered to be ‘on the Web’. According to Palmer (2001), a resource is anything that has an identity and that can be referenced by using a Web identifier such as a URI. Furthermore, the use of URIs as a global identification mechanism is what makes the Semantic Web possible. The Uniform Resource Locator (URL), which is a subset of the URI, specifically identifies resources by using their network locations (Berners-Lee & Masinter, 1994).

The Internet Engineering Task Force (IETF, 2003) governs the syntax and specification of URI. The general URI specification by the IETF is known as RFC 2396. Both URL and URI are accepted Internet standards (Berners-Lee, 2005; IETF, 2006).

XML, XML Schema and Namespaces

XML, XML Schema and Namespaces reside in layer 2 of the Semantic Web architecture (Berners-Lee et al., 2001; Berners-Lee, 2003). The function of the layer 2 technologies of the Semantic Web is to provide a self-describing syntax for the upper layer language technologies. In other words, it provides a syntax description mechanism for data interoperability (Gerber, 2007).

XML specifies a standard for the exchange of data over the Web (Bray et al., 1999, 2004; Palmer, 2001). XML as a mark-up language allows the insertion of mark-up tags into text to define the logical structure of a document or to add information regarding information contained in a document, i.e. metadata (Gerber et al., 2006). XML as a standard for data exchange over the Web is crucial for the enhancement of interoperability of the Web, as any XML parser can parse the XML data and access the content if it is a valid XML document (Bray et al., 2004).

XML Schema defines the contents and structure of XML documents (W3C, 2001a). XML Schema is a content-modelling language that describes the possible arrangements of elements, their attributes and text in a schema-valid document (Decker, Mitra & Melnik, 2000b; W3C, 2001a).

According to Bray, Hollander, & Layman (1999), an XML namespace is a collection of names, identified by a URI reference (RFC2396), which are used in XML documents as element types and attribute names. In other words, Namespaces (NS) provides a simple method for qualifying element and attribute names used in XML documents.

These technologies, i.e. XML, XML Schema and Namespaces, can be used to encode anything that has a defined grammar (Decker et al., 2000b).

The W3C released the second edition of Namespaces in 2006 (Bray, Hollander, Layman & Tobin, 2006a). XML Schema was endorsed in 2001 (W3C, 2001a; 2001b), and the fourth edition of XML was released in 2006 (Bray et al., 2006b).

RDF and RDF Schema

RDF and RDF Schema reside in layer 3 of the Semantic Web architecture (Berners-Lee et al., 2001; Berners-Lee, 2003). The function of the layer 3 technologies of the Semantic Web is to provide a metadata description mechanism for the upper-layer language technologies (Gerber, 2007).

RDF provides a mechanism for declaring statements that describe resources by means of a basic data model (Bray et al., 1999; Palmer, 2001). RDF is a vital language in realising the objectives of the Semantic Web as it is used to declare metadata that are machine-processable. Moreover, RDF enhances semantic interoperability because of the data model used (W3C, 1999; Decker, Melnik., Van Harmelen, Fensel, Klein, Broekstra, Erdmann & Horrocks, 2000a; W3C, 2004b).

RDF Schema extends RDF by defining common vocabularies in RDF metadata statements. RDF Schema is used to provide application-specific classes and properties. It also assigns externally specified semantics to specific resources (Bray et al., 1999; Berners-Lee et al., 2001; Palmer, 2001; Horrocks & Patel-Schneider, 2003). The objectives of the RDF Schema (the RDF vocabulary language) are to describe properties and to provide mechanisms for describing relationships between properties and resources (W3C, 1999; W3C, 2004c).

The W3C RDF Core Group has submitted various recommendations including the RDF Primer (W3C, 2004b), which is an introduction to RDF and RDF Schema, and a tutorial on how to use RDF and RDF Schema. The RDF Vocabulary Description Language 1.0: RDF Schema (W3C, 2004c) explains how to use RDF to describe application- and domain-specific vocabularies. These recommendations form a part of the W3C recommendations released in February 2004 (W3C, 2004d).

Ontology Vocabulary/Ontology and Rules

Ontology is a shared, formal, explicit specification of a particular domain (Decker et al., 2000b). It specifies a machine-readable vocabulary in computer system technology descriptions. Ontologies are used to specify and manage concepts, attributes and relationships between concepts (Bussler, Fensel, & Maedche, 2002). Ontologies are important in processing, sharing, and reuse of knowledge between Web applications (Decker et al., 2000b). *Rules* are pieces of declarative knowledge, imperative in managing complex and dynamic operations (Hitzler, Angele, Motik & Studer, 2005).

Ontology Vocabulary/Ontology and Rules resides in layer 4 of the Semantic Web architecture (Berners-Lee et al., 2001; Berners-Lee, 2003). The function of layer 4 of the Semantic Web architecture is to provide technologies that establish common knowledge representation formalism and a common understanding of domain concepts on the Semantic Web (Gerber, 2007).

OWL is a W3C technology for *Ontology* (Bechhofer, Van Harmelen, Hendler, Horrocks, McGuinness, Patel-Schneider & Stein, 2004), whereas efforts are still being made to establish technologies for *Rules* (Horrocks et al., 2005). Such efforts include the W3C RIF (Rule Interchange Format) and the SWRL (Semantic Web Rule Language). The RIF allows for *Rules* extension above the RDF-Schema. The SWRL is a combination of the decidable subset of OWL and the Rule Mark-up Language (Horrocks et al., 2004). SWRL has the ability to provide support for complex relationships between properties, thereby extending the expressiveness of what can be defined in OWL-DL.

OWL provides a knowledge representation language for capturing the syntax and semantics of a specific domain. It facilitates greater machine interpretability of Web contents than do XML and RDF by providing more vocabulary along with formal semantics (McGuinness & Van Harmelen, 2004). In other words, OWL has a higher ability to represent machine-interpretable information on the Web. OWL also possesses computational properties that enable reasoning tasks to be performed by machines, which is an essential feature of the Semantic Web.

OWL has three increasingly expressive sublanguages, namely, OWL Lite, OWL-DL, and OWL Full. *OWL Lite* supports a classification hierarchy and simple constraints. *OWL-DL* supports maximum expressiveness while retaining computational completeness and decidability. *OWL Full* supports maximum expressiveness and syntactic freedom of RDF with no computational guarantees (McGuinness & Van Harmelen, 2004).

The OWL specification was endorsed as a W3C Recommendation in February 2004 (McGuinness, 2004; Smith, Welty & McGuinness, 2004; W3C, 2004d). The latest version of OWL is OWL 1.1 and it was adopted as a W3C Recommendation in May 2007. OWL 1.1 replaces the three sublanguages. No formal submission has been made for

Rules by the W3C RIF working group (W3CRule, 2005), which was formed to assist in the establishment of rule language (Horrocks et al., 2004).

The emerging functionalities

The three top layers of the Semantic Web architecture are referred to as emerging functionalities because the functionality of the layer rather than the technology is mentioned (Gerber et al., 2006).

Logic Framework

The Logic framework resides in layer 5 of the Semantic Web architecture (Berners-Lee, 2003). The function of the logic framework is to provide a logic language on top of the ontology language that allows additional mechanisms for reasoning about formalism and integration of logic languages (Gerber, 2007). A logic framework provides formal semantics, which assigns unambiguous meaning to logic statements, which is necessary for inferencing on the Semantic Web (Decker et al., 2000a; McGuinness, Fikes, Hendler, & Stein, 2002).

Proof

Proof resides in layer 6 of the Semantic Web architecture (Berners-Lee et al., 2001; Berners-Lee, 2003). The function of the proof is to provide a mechanism to be used to determine the validity of a specific statement (Gerber, 2007). In the Semantic Web, proof languages (lists of inference items) are used to determine the validity of information together with the associated trust information of each item (Palmer, 2001).

Trust

Trust resides in layer 7 of the Semantic Web architecture (Berners-Lee et al., 2001; Berners-Lee, 2003). The function of trust is to provide mechanisms for establishing trust levels of information items and all entities that interact with the Semantic Web (Palmer, 2001; Gerber, 2007). On the Semantic Web, the context of information will assist applications and users of information with aspects relating to trustworthiness and usefulness of information (Thuraisingham, 2002).

The vertical layers

Signature and Encryption

According to Berners-Lee (2003), the Semantic Web architecture also includes two vertical layers, namely, Signature and Encryption. The function of these vertical layers is to provide security mechanisms that support the language architecture (Gerber, 2007).

Signature, which is enforced by XML Digital Signature (Bartel et al., 2002; Klyne, 2002), is a mechanism used to sign and verify entities unambiguously. It may be used for authenticity verification for retrieved and/or updated information, agents involved, etc. (Park, 2003; Horrocks et al., 2005). The use of XML Digital Signature in the Semantic Web results in a system that can express and reason about relationships among public key-based security and trust systems.

Encryption, which is enforced by XML Encryption (Imamura et al., 2002; Klyne, 2002), supports end-to-end encryption of an XML object, which can be the whole or a part of an XML document. It is an effective way to achieve data security (Gerber et al., 2006). It may be used for information storage, internal/external information transfer, as well as authentication (Park, 2003; Horrocks et al., 2005).

XMLDSig is a joint IETF/W3C standard for digitally signing and verifying a signature of an XML data object (Bartel et al., 2002). XMLEnc is a W3C standard for encryption and decryption of XML-formatted data objects (Imamura et al., 2002).

Table 2.1 below summarises the Semantic Web architecture in terms of its layers, functionalities and technologies.

Table 2.1: Summary of the Semantic Web architecture status model

Layer	Functionality	Technologies
Layer 1	Unique Identification	Unicode and URI
Layer 2	Syntax Description Language	XML, XML-Schema, and Namespaces
Layer 3a	Metadata Data Modelling	RDF
Layer 3b and 4a	Ontology	RDF-Schema and OWL
Layer 4b	Rules	
Layer 5	Logic Framework	
Layer 6	Proof	
Layer 7	Trust	
Vertical Layers	Security mechanisms	XMLDSig and XMLEnc

2.2.3. Concluding remarks

Within this section, an overview of the Semantic Web was presented. The overview of the Semantic Web included a brief background on the Semantic Web followed by the definition and use of the Semantic Web. The layered architecture of the Semantic Web was presented, together with associated Semantic Web technologies.

The overview of the Semantic Web provides a basis from which security aspects of the Semantic Web are discussed in the following section.

2.3. SECURITY ASPECTS OF THE SEMANTIC WEB

2.3.1. Introduction

The autonomous, dynamic, and heterogeneous nature of the Semantic Web entities brings new security challenges to the deployment of Semantic Web applications. In order to establish security features and functionalities that are desirable for a security framework for the

Semantic Web, a discussion of security aspects related to the Semantic Web is presented in this section.

The goal of security is to attain confidentiality, integrity and availability of computing *assets* such as information, applications, etc. through the use of *controls* such as authentication, authorisation, audit, and so on (Wallace, 2002; Pfleeger & Pfleeger, 2003). A control is an action, device, procedure or technique that removes or reduces a *threat* such as interruption, interception, modification or fabrication of computing resources.

In order to provide security for a computing system, it is necessary to identify computing assets to be protected, determine threats to the assets, and evaluate controls to achieve the desired security (Pfleeger & Pfleeger, 2003). In this section, the computing assets on the Semantic Web that need protection are discussed first. This is followed by a discussion of the security threats to the protected assets. This section will conclude with a discussion of the desired security services for the Semantic Web.

2.3.2. Semantic Web assets that need to be protected

In the context of Web applications, data and computational services are the assets that need to be secured (Li & Pahl, 2003; Pfleeger & Pfleeger, 2003). Furthermore, Kagal et al. (2003) identified agents, Web services and Web resources as assets that need protection on the Semantic Web. According to Kagal et al. (2003), it is necessary to describe the security functionality of the three main categories of entities prevalent on the Semantic Web, i.e. Web services, agents and Web resources.

In general, entities refer to objects that are targets of actions in an interaction, for example, computing resources (Uszok et al., 2004a). Semantic Web entities, specifically, are those objects that interact with the Semantic Web. In the Semantic Web, entities such as agents, Web services and Web resources participate in different kinds of interactions (Finin & Joshi, 2002). In the remainder of this section the Semantic Web assets that need to be protected are discussed.

2.3.2.1. Agents

A software agent is an autonomous software entity that can interact with its environment (OMG, 2000). Software agents are associated with several attributes including autonomy, interactivity, adaptivity, mobility, proactivity, coordinativeness, and cooperativeness. Mobile agents are used to represent a user on the network by roaming among Web services and other agents performing computational or other tasks on behalf of a user. Stationary agents are used to provide support and services to other agents to facilitate the completion of a task (Karnik, 2000).

The word *agent* has been used in a variety of contexts, ranging from robotics to networking to artificial intelligence to human-computer interaction to distributed systems. Other systems that are associated with the term agent include intelligent routers, Web searching tools, e-commerce applications, robots and many more.

Agent technologies were designed with a focus on interoperability, distributed problem solving and cooperation. Multi-agent systems were intended to be responsive to open environments such as the Internet to capitalise on cooperative interactions (Farkas & Huhns, 2002). Owing to the limited scope required for this dissertation, the discussion of agent technologies is excluded from this study. However, agents as assets to be protected and agent functionality are included.

On the Semantic Web, it is envisioned that a user would have a personal agent that would solve problems related to information overload, acquisition and discrepancy resolution (Decker et al., 2000a). Agents will assist a user by performing complex information management tasks on the user’s behalf. The increased semantic interoperability provided by the Semantic Web will enable agents to search and collect Web contents from different sources, process the information and exchange the results with other programs on behalf of its users. Semantic Web technologies such as RDF and OWL often facilitate agent interaction on the Web (Farkas & Huhns, 2002).

As agents become the eventual users of the Semantic Web, their interactions with other Web entities need to be explored. Security issues associated with agents include protecting hosts against malicious agents, protecting agents against malicious hosts, and protecting the network communications (Vuong & Fu, 2002). According to Claessens, Preneel and Vandewalle (2001), agents should be protected while they are in transit from one host to another. The communication between agents and users, and between agents themselves, should also be protected from malicious agents, hosts and users, as well as other entities. These entities could potentially eavesdrop on, or tamper with, the communication, or impersonate participating entities.

2.3.2.2. Web services

Web services are Web-accessible programs and devices (Ankolekar et al., 2001). According to the W3C, a Web service is ‘a software application identified by a URI (Uniform Resource Identifier), whose interfaces and bindings are capable of being defined, described and discovered by XML (Extensible Mark-up Language) artefacts. A Web service supports direct interactions with other software agents using XML-messages exchanged via Internet-based protocols’.

The proposed implementation of Web services on the Web to describe and compose services provides a framework for the implementation of distributed computing over the Internet (Li & Pahl, 2003). This framework addresses the interoperability problems in a heterogeneous distributed system and supports the concept of discovering services offered by other software components.

On the Semantic Web, Web services utilise Semantic Web technologies to facilitate automatic description, publishing and discovery of services, service flow, and composition. In Semantic Web Enabled Web Services (SWWS), ontologies are used to annotate Web services with machine processable metadata (Denker et al., 2003). Semantic Web-enabled Web Services will enable automatic discovery, selection and execution of inter-organisational business logics (Bussler et al., 2002). Web services discovery and composition is one of the main areas of Ontology use within OWL (Heflin, 2004).

According to Li and Pahl (2003), the security issues associated with Web services include the description of security requirements and constraints for the application of Web services, retrieving Web services from repositories that match client security requirements, and implementing security requirements when services are invoked across the Web.

2.3.2.3. Web resources

A Web resource may be an entire Web page, a part of a Web page, a whole collection of pages, or an object that is not directly accessible via the Web, for example, a printed book (Patel-Schneider & Fensel, 2002). According to Palmer (2001), a resource is anything that has an identity and that can be referenced by using a Web identifier such as a URI. Web resources are always named by URIs or URLs (Patel-Schneider & Fensel, 2002). The URL specifically identifies resources by using their network locations (Berners-Lee & Masinter, 1994). Users and agents may request different kinds of access to Web resources.

On the Semantic Web, resources are annotated with information about resources, i.e. metadata. Metadata is machine-understandable information about Web resources (Berners-Lee, 1997). The W3C recommended RDF as a language for representing metadata or information about Web resources (W3C, 2004a). RDF describes Web resources in terms of their properties and property values. RDF will be used for the exchange of metadata about resources between applications without loss of meaning (Decker et al., 2000a). It provides a mechanism for integrating diverse sources of information. According to Kagal et al. (2003), security issues associated with Web resources include authentication and fine-grained access control of resources.

The assets described in Section 2.3.2 above, namely, agents, Web services and Web resources, need to be included in a security framework for the Semantic Web, as they participate in different kinds of interactions with the Semantic Web (Finin & Joshi, 2002; Denker et al., 2003). However, it should be noted that the Semantic Web itself is a collection of Web resources of which agents and Web services are its main users. Figure 2.6 below illustrates the Semantic Web assets that need protection.

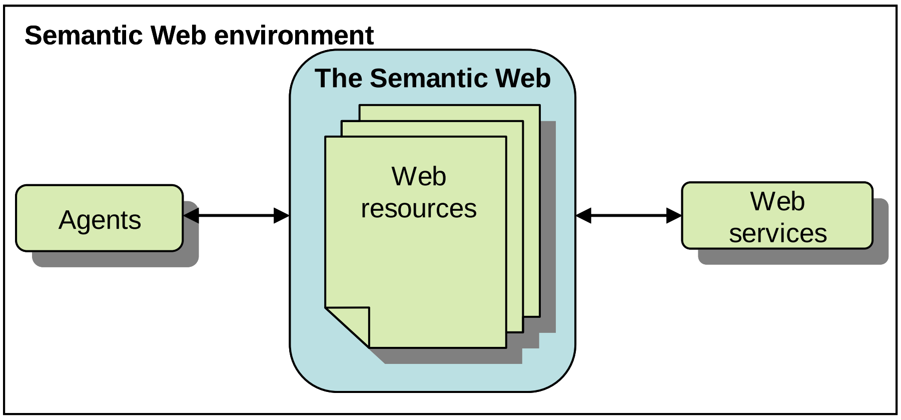


Figure 2.6: Semantic Web assets that need protection

2.3.3. Security threats

According to Pfleeger and Pfleeger (2003:6), a *security threat* to a computing system is a set of circumstances that has the potential to cause loss or harm. Security threats are categorised into interceptions, interruptions, modifications and fabrications (Pfleeger & Pfleeger, 2003:7).

2.3.3.1. Interception

Interception refers to unauthorised access of computing assets (Pfleeger & Pfleeger, 2003:7). In the Semantic Web, computing assets such as Web services, Web resources and software agents are vulnerable to interception.

In Web services an interception occurs when a ***session hijacking*** attack is executed. Session hijacking is an interception where a third entity intercepts and carries on a session begun by other entities (Pfleeger & Pfleeger, 2003:408).

In using Web resources, an interception may occur in transit, such as ***eavesdropping*** and ***passive wire-tapping***, or at the host by ***impersonation***, ***theft, spoofing,*** and ***illegal inferences***. Eavesdropping occurs when an intruder monitors traffic passing through a node (Pfleeger & Pfleeger, 2003:398). In passive wire-tapping, an intruder monitors the traffic through some efforts (Pfleeger & Pfleeger, 2003:398). Impersonation occurs when an entity pretends to be another entity. Impersonation is achieved through foiling of the authentication mechanisms (Pfleeger & Pfleeger, 2003:404). Theft occurs when an unauthorised entity obtains Web resources illegally such as copying of data and pirating software applications (Pfleeger & Pfleeger, 2003:15). Spoofing refers to obtaining authentication credentials of an entity (Pfleeger & Pfleeger, 2003:407). Illegal inference occurs when an entity derives sensitive information from non-sensitive data (Pfleeger & Pfleeger, 2003:331).

Software agents are also vulnerable to interception, in particular eavesdropping, where a malicious host may spy on the agent’s data and gather information about intercommunication between agents (Jansen, 2000).

Interception threats affect the *confidentiality* of computing assets (Pfleeger & Pfleeger, 2003:36). To preserve the confidentiality of

Semantic Web assets, there is a need for controls to overcome the interception threats discussed above.

2.3.3.2. Modification

Modification occurs when an unauthorised entity tampers with an asset (Pfleeger & Pfleeger, 2003:8). In the Semantic Web, Web services, Web resources and software agents are vulnerable to modification threats.

Active wire-tapping is a modification threat to Web resources and Web services. In active wire-tapping, an intruder monitors the traffic and injects something into the traffic (Pfleeger & Pfleeger, 2003:398). On the Semantic Web, Web resources requested by agents may be intercepted and modified by malicious hosts while in transit.

For the software agents, a host may perform an ***information modification*** attack by interfering in interactions between agents and altering the communication between them for its own benefit (Jansen, 2000).

Modification threats affect the *integrity* of computing assets (Pfleeger & Pfleeger, 2003:36). The integrity of Semantic Web assets needs to be preserved, hence the need to have controls for modification threats.

2.3.3.3. Interruption

Interruption refers to attacks on computing assets that result in lost, unavailable or unusable assets (Pfleeger & Pfleeger, 2003:7). In the Semantic Web, interruption may occur on Web services, Web resources and software agents. Interruption on these entities may be done by ***denial of service*** or intentional ***deletion*** of Web resources. Denial of service is an availability attack that could be accidental or malicious (Pfleeger & Pfleeger, 2003:414). Denial of service is mainly caused by transmission failure, connection flooding, and traffic redirection.

A Web service may be interrupted by a denial-of-service attack so as to stop it from providing a certain service. Moreover, a Web service may be flooded with so many requests to provide a service that it cannot receive any more data.

Deletion and loss of Web resources such as data files are regarded as an interruption threat. In the Semantic Web, Web resources could be deleted by viruses and thus cause unavailability of Web resources.

Software agents also may be attacked by a denial of service, where resources requested by an agent to complete its mission are denied (Jansen, 2000).

Interruption threats affect the *availability* of computing assets (Pfleeger & Pfleeger, 2003:36). The availability of Semantic Web assets needs to be preserved for its users to benefit from it.

2.3.3.4. Fabrication

Fabrication refers to the creation of counterfeit objects on a computing system (Pfleeger & Pfleeger, 2003:8). In the Semantic Web, it is possible to fabricate Web Services, Web resources and software agents.

Fabrication of Web services and Web resources is performed through ***phishing,*** where attackers use spoofed emails and fake websites to obtain users' credentials (Knight, 2004). In ***masquerading*** the attacker conceals his or her true identity and uses the identity of a trusted valid entity (Pfleeger & Pfleeger, 2003:407). In Web services, an intruder may insert spurious transactions that affect other entities.

Cloning is a fabrication threat on a software agent, where a host may create a clone of an agent in order to gain unauthorised access to the services of the agent's executing host (Jansen, 2000).

Fabrication threats affect the *integrity* of computing assets (Pfleeger & Pfleeger, 2003:36). Fabrication threats on the Semantic Web may destroy the integrity of Semantic Web assets, hence a need for controls to avoid the fabrication threats discussed above.

2.3.4. Security services

Security services that can be provided in a network environment include encryption, authentication, authorisation, confidentiality, integrity, availability, non-repudiation and audit (Wallace, 2002; Pfleeger & Pfleeger, 2003; Detsch, Gaspary, Barcellos, & Cavalheiro, 2004;).

Essential security services for Web systems are derived from the requirements of users gaining access to computer network services set by the ISO (ISO 7498-2, 1988), which include authentication, authorisation, integrity, confidentiality, non-repudiation and availability.

Audit is the recording of activities that took place on a computing system, including who accessed what, when and for what amount of time (Pfleeger & Pfleeger, 2003). Audit service is normally performed by the Web server or by the network management system. Audit service entails issues such as privacy of interacting parties, which have not yet been resolved.

2.3.4.1. Authentication

Authentication is the process of verifying that an entity (a person, device, application, network, or agent) is indeed who it says it is (Pfleeger & Pfleeger, 2003). Authentication forms a basis for security in computing as most other security services depend on the authenticity of a subject in question. For instance, an authorisation process might allow anyone claiming to be user U_1 to access an object O_1. If the authentication of user U_1 is compromised, the authorisation process will also be compromised.

2.3.4.2. Authorisation

Authorisation is the mechanism for deciding which subjects should have access to which objects (Pfleeger & Pfleeger, 2003). The process of ensuring that only authorised subjects will be able to access only authorised objects is done through implementing access controls, permissions, privileges and other elements, depending on the systems involved (Wallace, 2002).

In the Semantic Web, authorisation is required to not only access services and data but also their metadata. In some instances the actual resources and their metadata can be contained in the same file, such as an XML file, whereas in other instances resources and metadata can be in different files linked by URIs (Park, 2003). In either case it is imperative to provide access control to both the metadata and the resources.

2.3.4.3. Integrity

Integrity means that assets can be modified only by authorised parties or only in authorised ways (Pfleeger & Pfleeger, 2003). In the context of the Semantic Web, integrity could mean Web resources are modified only in acceptable ways, modified only by authorised people, modified only by authorised services and internally consistent. Based on the above definition, integrity is therefore dependent on authentication and authorisation processes. Cryptographic technologies such as encryption and digital signature are usually used to enforce integrity (Park, 2003).

2.3.4.4. Confidentiality

Confidentiality means that assets are accessed only by authorised parties (Pfleeger & Pfleeger, 2003). In other words, confidentiality means that Web resources (including data) can be read, viewed, printed or known to exist by authorised parties only. Confidentiality, which is also known as *privacy,* is commonly enforced by the use of encryption methods.

The inference problem (deriving confidential data from non-confidential data) is a confidentiality problem on the Semantic Web (Farkas & Huhns, 2002). With data-mining tools one can make all types of inferences on the Semantic Web. Inference is a desirable feature of the Semantic Web, but it should be controllable in such a way that one should only be able to make inferences where it is permitted. Inference impacts on the confidentiality and privacy of Web resources. Issues relating to inference problems are still being investigated.

2.3.4.5. Availability

Availability means that assets are accessible to authorised parties at appropriate times (Pfleeger & Pfleeger, 2003:10). Availability applies to both Web resources and Web services.

Denial-of-service is a common attack in respect of availability, however, even presentation of data and services in an unusable form results in availability problems (Pfleeger & Pfleeger, 2003:12). In the context of the Semantic Web, for instance, the issue of interoperability could result in non-availability of Web services or Web resources. For instance, if Web resources are described in a specific language that is not universal, other agents that need to use those Web resources may not be able to access or process them.

2.3.4.6. Non-repudiation

Non-repudiation means that senders or clients cannot deny having sent a message or performed a transaction (Pfleeger & Pfleeger, 2003:474). Digital signatures are used to enforce non-repudiation, whereby a party’s signature is attached to a message or transaction and can be saved by the receiver of the message for future disputes.

Since agents will be carrying out transactions with different Web services such as booking air tickets, ordering items, etc. on the Semantic Web, these agents should not be able to refute these transactions later on. Semantic Web systems should therefore ensure that transactions are properly traceable and accountable to authenticated individuals and that they cannot subsequently be disavowed (Claessens, Preneel & Vandewalle, 2003).

The degree to which these security services are needed varies from one application domain to another (Kemmerer, 1998). For instance, the defence industry might consider confidentiality more important, whereas the banking industry might consider integrity as a priority security service.

2.3.5. Concluding remarks

Since the Semantic Web is the extension of the World Wide Web, which is a Web of inter-networks, it inherits security threats from both networks and the Internet. The security threats inherited from networks include data communication threats such as eavesdropping, wire-tapping and session hijacking. The security threats inherited from the Internet include attacks such as impersonation, spoofing, phishing, masquerading and denial-of-service.

The completely decentralised nature of the Semantic Web, the extremely large number of users, agents and services and their heterogeneity makes security increasingly difficult to achieve. The autonomous nature of the Semantic Web, which supports complex and dynamic relationships between clients and service providers, brings new security challenges such as illegal inference, access control of Web resources, and metadata security.

In order to summarise these threats, each protected asset is grouped with its associated security threat and the security goal affected by the threat. Table 2.2 below integrates security aspects related to the Semantic Web including protected assets, security threats, and security goals that are affected.

Table 2.2: Security aspects related to the Semantic Web

Protected asset	Security threat	Security goal affected
Web services	Session hijacking	Confidentiality and integrity
	Eavesdropping	Confidentiality
	Wire-tapping	Confidentiality and integrity
	Impersonation	Confidentiality and integrity
	Spoofing	Confidentiality
	Phishing	Confidentiality, integrity and non-repudiation
	Masquerading	Non-repudiation
	Denial-of-service	Availability
Web resources	Eavesdropping	Confidentiality
	Wire-tapping	Confidentiality and integrity
	Impersonation	Confidentiality and integrity
	Denial-of-service	Availability
	Deletion of resource	Availability
	Phishing	Confidentiality, integrity and non-repudiation
	Illegal inference	Confidentiality
Agents	Information modification	Integrity
	Masquerading	Non-repudiation
	Cloning	Confidentiality and integrity
	Denial-of-service	Availability
	Eavesdropping	Confidentiality

EXISTING SECURITY FRAMEWORKS

2.4.1. Introduction

In order to compile a security framework for the Semantic Web, it is essential to study existing security frameworks and to establish their applicability to the Semantic Web.

Studying the existing security frameworks will assist in justifying the need to establish a new security framework for the Semantic Web. Understanding how existing security frameworks were compiled will assist in this study in establishing similar or better methods of compiling a security framework and deciding on features and components to be included in the framework.

According to Alter (1996), a framework is a brief set of ideas for organising a thought process about a particular type of thing. The Pocket Oxford Dictionary defines *framework* as “an essential supporting structure” (POD, 1994). Other terminologies related to the term *framework* include *infrastructure*, *architecture*, and *model* (Bass, Clements & Kazman, 2003; Avison & Fitzgerald, 2006).

Infrastructure is defined as a basic structural foundation of a system (POD, 1994). *Architecture* is defined as a structure of a system within a specific context (Bass et al., 2003). A *model* is an abstraction representing a proposed structure (Avison & Fitzgerald, 2006). The commonality in the above-mentioned terminologies is the presence of a structure or organisation of some ideas for a particular thing. For the purpose of this study, the term *framework* is therefore defined as **an approach that presents a structure or organisation of concepts to support a system with a specific goal**. In this case, the goal is the security of the Semantic Web. This framework description may include an abstraction or model of such a structure. In addition, the framework may include an organisation of system components, also defined as system architecture.

The discussion of a particular existing security framework below will be structured as follows: (1) a brief description of the framework including the objectives of the framework, (2) a brief description of its components and security functionalities, and (3) an evaluation of the framework in relation to this study.

2.4.2. Summary of existing security frameworks

In this section security frameworks are studied from literature. For the purpose of the discussion, the frameworks are categorised in order to indicate the context of their application. This categorisation of the security frameworks is based on the development process of the Web as presented by Fensel (2002). According to Fensel (2002), the Web development process started from a static Web popularly known as the

World Wide Web (WWW) that uses technologies such as URL, hypertext transfer protocol (HTTP), hypertext mark-up language (HTML), etc. The World Wide Web then evolved to a dynamic Web popularly known as ***Web services*** that uses technologies such as Universal Description, Discovery, and Integration (UDDI), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), etc. Meanwhile the Web service is evolving towards a machine-understandable Web, known as the ***Semantic Web*** that uses technologies such as RDF-S and OWL. The evolution of the Web dictated the evolution of security mechanisms and approaches that were developed to tackle new challenges.

The security frameworks discussed are therefore grouped into four categories.

- The first category involves security frameworks that were designed for general Web applications. This category includes distributed systems such as enterprise middleware systems, peer-to-peer systems, grid computing systems, and mobile agent systems.

- The second category includes frameworks that were designed for Web services. This category involves Web services applications that utilise technologies such as UDDI or SOAP.

- The third category involves security frameworks that were designed specifically for the Semantic Web. These frameworks utilise Semantic Web technologies such as RDF-S or OWL in implementing the framework.

- The last category involves the XML-based security standards that have been adapted to provide security to the Semantic Web. The XML-based representations are easily convertible to richer semantic notations, hence adaptable to semantic services. The XML-based security standards are not security frameworks as such, but they make an important contribution towards providing security functionalities in distributed environments.

Within the categories discussed above, the security frameworks are arranged in a chronological order from the oldest to the most recent one to indicate the evolution of security frameworks.

2.4.2.1. Security frameworks for general Web applications

The digital distributed system security architecture

Gasser (1989) presented security architecture for digital distributed systems. The architecture is a comprehensive specification of security in distributed systems. The architecture covers issues such as user and system authentication, mandatory and discretionary security, secure initialisation and loading, and delegation.

The architecture is made up of distributed security policy, reference monitors, message authentication and secure channels.

- In the *distributed security policy*, each system implements its own reference monitor to enforce its own policies.
- *Reference monitors* control access to objects they maintains. Some level of mutual trust between reference monitors is needed to allow subjects from other reference monitors to access objects in other reference monitors.
- *Message authentication* is achieved by using message hash functions that yield Message Authentication Code (MAC), e.g. X.509.
- *Secure message channels* are transport layer connections that provide confidentiality and integrity of data. They may be defined by a given encryption key that is used to pass signed messages.

The digital distributed security architecture is regarded as an existing security framework because it presents a structure that supports security functionalities to general Web applications.

Security architecture for mobile intelligent agent systems

Vuong and Fu (2002) proposed a security architecture for mobile intelligent systems that provides a secure execution environment for both agents and hosts. The architecture supports decoupling of application functionalities with security processes such as authentication, access control and secure communications. The architecture is scalable, flexible and supports potential dynamic and open-ended growth in size and number of agents that need to interact in large-scale distributed systems such as grid computing systems. The architecture provides identification capability to each *principal*, and supports system resources access control to a very fine level of granularity. A principal is an entity whose identity can be authenticated. Each principal is associated with a certificate that provides the principal's privilege, role and a public key.

The main components of the architecture include host protection, agent protection and protecting communications.

- *Host protection*: This involves an authentication process that uses digital certificates and signatures for authentication and a secure execution environment to authorise access to resources. Host protection also includes an authorisation process that specifies and controls the extent to which an agent with a certain identity can use the agent platform's resources and services. Security policies are used to apply the access rules.
- *Agent protection*: This involves protection of code and data integrity of an agent from malicious hosts and other agents. An agent syntactic integrity check mechanism is used to detect if the agent's behaviour is tempered. An approach called 'append data log only' is used to prevent an agent's collected data from being tampered with.

- *Protecting communication*: This is achieved through setting up secure communication channels between agent platforms. SSL is used to provide encrypted communication that prevents eavesdropping attacks. It also provides mutual authentication to both sides of communication to prevent man-in-the-middle attack.

The security architecture for mobile intelligent agent systems is regarded as an existing security framework because it consists of organised concepts that support a secure execution environment for mobile agent systems.

Policy-based security framework for Web-enabled applications

Ventuneac, Coffey and Salomie (2003) proposed a flexible and adaptive policy-based security framework for web-enabled applications. The framework enables the implementation of security services in a modular approach. Furthermore, event-based user auditing, error handling and awareness auditing are supported by the framework.

The framework is made up of two main components, namely, the security standards and mechanisms, and a set of flexible security policies. The security standards and mechanisms define specific instances of security objects such as PKI, X.509 etc.

The security policies included are dynamic adaptive authentication policy, access control policy, security administration policy and accountability policy.

- The *authentication policy* defines which security mechanisms are used in specific identification contexts, based on the user's credentials.
- The *access control policy* specifies which entities are to be protected, against whom, and the security mechanisms to protect them.

- The *security administration policy* defines rules for user identities and privilege management as well as resource management.
- The *accountability policy* defines the system security auditing levels.

The policy-based security framework for Web-enabled applications complies with the definition of a security framework as defined in this study. It presents a structure (made up of the two main components) that supports security functionalities for Web applications.

Security framework for distributed brokering systems

Pallickara and Fox (2003) proposed a security framework for distributed brokering systems to ensure secure communication between authorised entities. The framework uses a topic-based publish/subscribe paradigm to address authentication, maintenance of identities, scalable topic security and message level security. The framework involves entities specifying an interest in a certain topic. The publisher will then publish messages to a given topic. Upon receipt of published messages, the system will compute the destination for the message. Every topic is associated with an Access Control List (ACL) identifying entities that are authorised to subscribe to messages published to that topic. A similar ACL exists for publishers.

The framework comprises the *broker network* and the *Key Management Centres* (KMC). The broker network comprises cooperating message nodes called *broker nodes* and the links between them. The KMC incorporates an authorisation module, which is used to keep track of authorisations that different entities within the system possess. The functions of the KMC include management of keys associated with entities and topics, registering entities' public keys, and ensuring secure communication with entities by using SSL. Brokers within the broker network are also involved in determining whether a publisher is indeed authorised to publish messages.

The framework secures messages independently of any transport level security in order to provide a fine-grained security structure for distributed systems and multiple security roles. Security services such as authentication are performed in a mechanism-independent way, with specific mechanisms mapped onto specific applications. Message-level security allows for deployment of secure communication links where data are not encrypted. The framework is capable of detecting security breaches by issuing authentication challenges at regular intervals along with the use of shorter key lifetimes. The framework responds to security breaches by generating new keys, propagating the detected breaches, and by encrypting replay of messages with new keys.

The policy-based security framework for Web-enabled applications is considered as an existing security framework because it is a structure made up of broker networks and KMCs that provide security functionalities to Web applications.

Flexible security framework for peer-to-peer grid computing

Detsch et al. (2004) proposed a flexible security framework for peer-to-peer based grid computing systems. The framework is intended to provide authentication, authorisation, integrity, confidentiality and audit services to Web services, Web resources and messages. The framework is modular and reconfigurable. The framework is based on JXTA (Juxtapose) and JAL (JXTA Abstract Layer).

The main components of the framework are the security profile and a configure module which lies between the application layer and the communication layer.

- The s*ecurity profile* groups a set of known peers that share common security requirements. It specifies peers that need a specific security service for specific tasks or interactions. For instance, a profile may specify that when peer 1 receives messages from peer 2 or peer 5, authentication service must be applied.

- The *configure module* is used to set the security requirements and the mechanisms to fulfil the security requirements specified on a security profile. It is responsible for discovering the peers that form a group specified in a profile.

Owing to its structure (consisting of security profiles and configure module), the flexible security framework for peer-to-peer grid computing is regarded as an existing security framework according to the compiled definition.

KAoS Policy and Domain Services Framework

Uszok et al. (2004a) proposed a policy and domain services framework for grid computing and Semantic Web services. KAoS is a collection of componentised services developed to increase assurance and trust with agents deployed in a variety of operational environments. The organisation of the components in KAoS provides a structure that supports security functionalities, hence its inclusion as a security framework.

KAoS domain services provides the capability for agents, users, resources and other entities to be semantically described and structured into domains and sub-domains to enhance collaboration and extend policy administration.

KAoS policy services allows for specification, management, conflict resolution and enforcement of policies within domains.

The functionalities of the KAoS framework are categorised into *generic functionalities* and *application-* and/or *platform -specific functionalities*. The main components of the KAoS Policy and Domain Framework are

- *Policy template* expresses authorisation or obligation for some type of action performed by one or more actors in a given situation.

- *Ontologies* define basic concepts for actions, actors, groups, places, entities, and policies.
- *Directory services* loads policy ontologies, including the structure of policies, domains, actors and other application entities, into the ontology repository.
- *Policy life-cycle management* provides extensive support for policy life-cycle management including a sophisticated policy disclosure interface for querying about policy impact on planned or executed actions.
- *Guards* provide the ability to register with KAoS services and check whether a given action is authorised or not based on current policies.
- *Enforcers* control, monitor and facilitate subclasses of actions.

The KAoS framework provides security capabilities to agents and Web services interacting on the Semantic Web and grid computing systems.

Security architecture for open collaborative environment

Demchenko, Gommans, De Laat, Oudenaarde, Tokmakoff, Snijders and Van Buuren (2005) proposed a security architecture for open collaborative applications that is flexible and customer-driven. The architecture integrates Web services and grid security technologies with generic AAA authorisation framework.

The main components of the architecture include

- *Communication security layer,* which defines network security infrastructures such as SSL, IPSec, VPN, etc.
- *Messaging security layer,* which uses WS-Security mechanisms and SAML for security token exchange format
- *Policy expression layer,* which defines sets of policies that can be used to entities that interact with the environment.
- *Services layer,* which defines security services for secure operations of the environment components. Security services

include authentication, identity management, authorisations, trust or secure context management, auditing and notarisation.

The security architecture for open collaborative environments is considered as a security framework according to the compiled definition of this study because it presents a structure that supports security functionalities for open collaborative applications.

2.4.2.2. Security frameworks for the Web services

Me-Services

Joshi, Finin and Yesha (2002) proposed a framework called Me-Services for secure and personalised discovery, composition and management of services in pervasive environments. The framework makes use of semantically rich profiles of agents and/or entities to enable multi-agent interactions.

The main components of the framework are semantic service discovery, service composition, profile-driven management, and distributed trust management.

- *The semantic service discovery* uses service description and the matching technique in discovering information and services. RDF-S/DAML-S is used to enable a reasoning engine to draw inferences from various service descriptions based on the ontology.
- *The service composition* creates new services by integrating and executing existing services in a planned manner. Dynamic service composition is described in a structured manner by languages such as WSDL or DAML-S.
- *The profile-driven management* manages the access, storage, monitoring, and manipulation of data and information based on context constraints. The context constraints such as location or user preferences are specified in profiles. Profiles are presented in DAML+OIL or DAML-S.

- *The distributed trust management* uses trust relationships as a way of authenticating entities and providing access control. In this approach, trust management is regarded as the establishment of trust relations rather than quantifying trust. Policies for user authentication, access control and delegation are specified in DAML.

Me-Services is regarded as a security framework because it provides a structure that uses Semantic Web technologies to provide security to Web services.

Security framework for Web services

Adams and Boeyen (2002) proposed a framework for providing security to Web services. The framework extends the existing UDDI (Universal Description, Discovery, and Integration) and WSDL (Web Services Description Language) as well as the security of the publish/discover mechanisms for Web services.

The main components of this framework include registry security, transaction security and infrastructure linkage.

- The *registry security* ensures that users of the registry feel confident that the information they retrieve from the registry is trustworthy. Trustworthy information is one that has authentic, authoritative, unmodified, confidential and current contents. Registry security involves UDDI registry operation security, UDDI stored data security, UDDI registry and node policy.
- The *transaction security* ensures that users of the Web service listed on the registry feel confident that the business transaction will be executed in a trustworthy manner. A trustworthy transaction is one that is authorised, unmodified, private, verifiable, non-repudiable, current, and credible. Transaction security involves trust attributes, facilitation of public key infrastructure, facilitation of trust policy infrastructure and facilitation of requester preferences.

- The *infrastructure linkage* ensures that all underlying infrastructures required for the trustworthiness of the UDDI environment can be understood and exploited by the participants involved. It involves authentication infrastructure, standard time infrastructure, and business infrastructure.

The security framework for Web services qualifies as an existing security framework since it provides a structure that support security functionalities.

Security for DAML Web services

Denker et al. (2003) proposed an approach that bridges the gap between the Semantic Web and security through security annotations (marked up in DAML) for agents and Web services at a very high abstraction level. In this framework, ontologies are used to describe the security requirements and capabilities of Web service providers and requesting agents.

The framework includes security annotations, a security reasoner (reasoning engine) and a semantic matchmaker.

- *Security annotations* express security-related capabilities and requirements of Web services and agents. DAML-S is used to describe Web services through a *service profile*, which describes high-level features of the Web service, a *service model*, which describes what the service does, and *service grounding*, which describes how to contact the service.
- *The security reasoner* accepts the requirements and capabilities of the agent and the service as input and decides to what degree they match.
- *The semantic matchmaker* searches for services that meet the functional requirements of the agent, then utilises the security reasoner to decide the subset of all discovered services that meet the security requirements of the requesting agent.

The framework provides security brokering between agents and services. Furthermore, the framework utilises Semantic Web technology to support security functionalities.

Semantic-based user privacy protection framework for Web services

Turner et al. (2005) proposed a security framework for Web services that allows agents to negotiate automatically with Web services on the amount of personal information to be disclosed on behalf of the user.

The main components of this framework are:

- *Policy statement*: used by the Web services to describe their business practices regarding the use of personal information. Policy statements are written in DAML-S.
- *Input requests*: these are data sets that users have requested as the input parameters of a Web service.
- *Privacy preferences* stores preferences of a particular user regarding the policy statement and input parameters of Web services.
- *The service request analyser* is responsible for parsing and analysing data request files and policy statements given by the service provider.
- *The rule extractor* is used to determine a user's privacy rules regarding a Web service to be utilised during negotiations between agents and the service.
- *The negotiation component* finds the ground for agreement between agents and Web services based on rules declaring the service's request and rules describing the privacy preference.

The framework provides a structure that supports the protection of privacy of information on Web services.

Semantic policy-based security framework for business processes

Huang (2005) proposed a security framework for business processes that governs the *orchestration* and *choreography* of business processes. Orchestration involves business logic and how Web services can interact, i.e., execution orders. Business process execution language (BPEL or WS-BPEL) can be used for orchestration. Choreography involves message exchange and is more collaborative in nature. Web service choreography description language (WS-CDL) is a W3C description language used for choreography. The framework provides security at two levels, namely *task level security* and *process level security*. Task level security is implemented by using the WS-Policy framework, whereas process level security uses Rei and SWRL.

The main components of the framework include:

- *Policies:* these specify security concerns, privacy, and business rules.
- *Meta BPEL* describes abstract processes with functional tasks.
- *Ontology repository* consists of both business ontology for business process and security ontology for security concepts.
- *Security services* enforce security requirements as specified on the security ontology and security policies.
- *Policy manager* carries out the policy matching, negotiations and conflict detection.
- *BPEL* integrates semantic policies, business rules and other security requirements.
- *BPEL engine* enforces the business process as specified on the BPEL.

The framework provides a structure that utilises Semantic Web technologies to support security functionalities for business processes.

Using semantic rules to determine access control

Shields, Molloy, Lyons and Duggan (2006) proposed a security framework for Web services that provides access control based on semantic rules. The framework extends the Web service security architecture on access control.

The framework suggests a semantically defined *Knowledge base* in OWL, semantically defined *Rules* in SWRL, the evaluation of rules, i.e. *reasoning engine* in OWL-DL, and *document filtering*, where unauthorised data are pruned from the data requested.

- The *knowledge base* is the description of information being protected that will be used in the authorisation process, and is defined in OWL.
- Semantically-aware *rules* written in SWRL are used to define access rights of entities to information represented in knowledge base.
- The OWL-DL *reasoning engine* is used to evaluate authorisation rules and generates authorisation decisions such as grant full access, grant limited access or deny any access.
- *Document filtering* examines document requests and authorisation decisions to prune unauthorised information before sending the response.

The framework provides a structure (knowledge base, rules, reasoning engine, and document filtering) that uses Semantic Web technologies (OWL and SWRL) to provide security functionalities to Web services.

2.4.2.3. Security frameworks for the Semantic Web

Concept-level access control for the Semantic Web

Qin and Atluri (2003) proposed an access control model for the Semantic Web that is used to specify access authorisations based on concepts and their relationships. Access authorisations are stated on concepts specified by ontologies.

The model consists of concepts and their relationships, propagation policies, authorisation conflict resolution, and a semantic access control language.

- A *concept* is defined as a set of *ontologies* in which the semantics of the concept is defined, *taxonomies* in which concepts are organised in a hierarchical structure, *properties, restrictions and values* of properties, and *instances* of a concept.
- *Relationships* defined by the model include superclass/subclass, equivalence, part/whole, overlap/intersection, sub-concept/union, and complement.
- *Propagation policies* allow propagations to be performed to extend authorisation to other concepts based on the relationships among concepts and propagation policies.
- *Authorisation conflict resolution* uses the explicit authorisation base and the propagated authorisation base to detect and resolve conflicts by creating a consistent authorisation base that does not contain any authorisation conflict.
- The *semantic access control language* uses the syntax and vocabulary of OWL to express concept-level access authorisations.

The model provides a concept-level access control for the Semantic Web by specifying authorisations over concepts defined in ontologies and enforcing them upon their data instances. It also supports the authorisation propagations based on the relationships among concepts. The model is regarded as a security framework as it provides a structure that supports security functionalities for the Semantic Web.

Semantic Web security infrastructure

Ashri et al. (2004) proposed a security infrastructure that uses Semantic Web technologies to improve security in service-oriented, open heterogeneous environments. The infrastructure makes use of conventional security solutions, together with the ability to reason about security at the semantic level, by using appropriate security policies. The infrastructure also makes use of a semantic firewall for the enforcement of security policies.

The main components of the Semantic Web security infrastructure are the description capabilities, reasoning capabilities, infrastructure capabilities and the semantic firewall.

- The *description capabilities* involve the capability of the infrastructure to describe security-related information such as context-dependent and context-independent security requirements, conventional and unconventional security requirements, as well as interaction scenarios.
- *Reasoning capabilities* involve identifying and resolving conflicts with site policies or interacting services' policies. The infrastructure is also able to reason about interaction processes.
- The *infrastructure capabilities* involve decoupling of security services, layered security support and context-dependent adaptation. The semantic firewall reasons about the acceptability of the incoming and outgoing messages based on the current context and the security policies in place.

The infrastructure contributes to this study by outlining requirements of a security framework for the Semantic Web. It also indicates the type of interactions that a security framework for the Semantic Web should be able to handle. The infrastructure complies with the definition of a security framework as it involves the organisation of concepts to support security for the Semantic Web.

Policy-based security approach for the Semantic Web

Kagal et al. (2003) proposed a security framework based on a policy language, which addresses security issues for Web resources, agents and Web services in the Semantic Web. The framework provides access control to entities without necessarily authenticating the requesters completely. The framework provides flexibility in specifying security requirements and gives every entity certain autonomy in making its own security decisions.

The framework consists of two main components, namely, the semantic policy language and distributed policy management.

- A s*emantic policy language* based on RDF-S, DAML+OIL or OWL is used to mark up security information. The policy language includes constructs, conflict resolution, speech acts and delegation management.
- In *distributed policy management,* every entity is capable of specifying its own security policies, which are enforced by a policy engine. The policy engine interprets and reasons about policies related to speech acts and domain information in order to make decisions about applicable rights, prohibitions, obligations and dispensations.

The framework is flexible and dynamically modifies existing policies. In this framework security is uniformly applied to and applicable to all Semantic Web entities.

Profile-based security model for the Semantic Web

Tan and Poslad (2004b) proposed a semantic model that supports policy-type constraints and a profile-based security information interchange for multi-domain services. Profiles are viewpoints of sets of safeguards that protect particular assets from particular security threats. A profile describes relationships among safeguards, assets and threats. Profiles can also express policy rules, defining security instantiations and preconditions supported. In this profile-based security model, conceptual representations of security entities are mapped onto explicit security specifications. Profiles can be constrained by policies. The model supports adaptive management, risk management and reasoning.

This layered security model consists of the following layers:

- *Security mechanisms layer:* has OASIS and W3C security specifications and specific instances of security concepts, policies and services. Security mechanisms are represented using DAML-OIL ontologies to enable entities to specify security requirements and capabilities.
- *Conceptual layer*: defines properties and relations among security, trust and privacy issues.
- *Reification layer*: has service descriptions, policies and trust sub-layers. In the service description layer security processes are hooked into service processes. Service descriptions are published in DAML-S, and possibly in Web service choreography interfaces (WSCI) and BPEL4WS+WSDL. The policy layer defines security rules and constraints. The trust layer defines trust implementations within the system.
- *Security applications layer*: has security management, policy management, and risk management sub-layers. This layer makes use of security ontology within a specific application domain.

The profile-based security model for the Semantic Web is regarded as a security framework as it provides a structure that supports security functionalities for the Semantic Web.

Using RDF for policy specification and enforcement

Carminati, Ferrari and Thuraisingham (2004) proposed a security framework that utilises the semantic richness of RDF for expressing security information and hence making policy specification and enforcement easier.

The main components of the framework include:

- *RDF description*: This is a set of RDF statements describing a scenario including security requirements.
- *Security enhanced RDF descriptions*: These are documents containing access control constraints in RDF and are used in generation of policies.
- *High-level policy generation*: This component generates high-level policies for a given RDF description of the general scenario and stores these policies on a *high-level policy base*.
- *Authorisations entailment* uses the RDF description of a scenario and the high-level policies to generate the corresponding authorisations.
- *Reference monitor* receives an access request as an input and checks whether it can be granted or not.

The framework is capable of automatically entailing all the authorisations implied by the application of the high-level policies to a specific scenario. The framework provides a structure that is suitable for specifying and enforcing security policies.

2.4.2.4. XML-based security standards

As explained earlier, XML-based security standards are not complete security frameworks. However, they contribute to such frameworks by providing security functionalities to be used when constructing security frameworks. The following sub-section discusses XML-based security standards.

XMLDSig

XML Digital Signature is a mechanism to sign and verify an entity unambiguously (Bartel et al., 2002). It is a method of associating a key with referenced data. It may be used for authenticity verification of retrieved information or verification of an entity requesting access to resources on the network environment.

XMLEnc

XML Encryption is a standard for encryption and decryption of XML data objects (Imamura et al., 2002). It supports end-to-end encryption of the XML objects. Encryption can be performed upon a whole document or part of the document, hence the potential for supporting a finely grained access control.

XKMS

XML Key Management Specification is a protocol for distributing and registering public keys (Ford et al., 2001). XKMS is a useful key verification mechanism for use with XML signatures. XKMS comprises two components, namely, the XML Key Information Service Specification (X-KISS) and the XML Key Registration Service Specification (X-KRSS).

- X-KISS defines a protocol that supports delegation by a service in processing of Key Information associated with an XML signature, XML encryption, or other public key. X-KISS is used to locate the required public keys and to describe the binding of such keys.
- X-KRSS defines a protocol for a Web service that accepts registration of public key information to be used in conjunction with other Web services.

WS-Security

Web Service Security (WSS) provides mechanisms to secure SOAP message exchanges (Klyne, 2002). WSS is designed to be extensible in order to accommodate a variety of authentication and authorisation mechanisms. WSS enhances SOAP messaging by providing quality protection through the application of message integrity, message confidentiality and single message authentication to SOAP messages.

The main components of this framework are SOAP security header, message security model, global identifier attribute, and signing and encrypting mechanisms.

- *The message security model* is defined in terms of security claims, endorsement of claims, and verifiable proof of possession.
- *The SOAP security header* is attached to a message and contains security-related information targeted at a specific receiver.
- *The global identifier attribute* is a simple way to identify specific XML content in a SOAP message.
- *The signing mechanism* uses XMLDSig when signing SOAP messages, whereas *the encrypting mechanism* uses XMLEnc when encrypting SOAP messages.

WSS provides a number of integrating mechanisms for authentication, privacy and authorisation in SOAP-based applications.

SAML

SAML is an XML-based framework for communicating user authentication, entitlement and attribute information (Klyne, 2002). In SAML a single user authentication is used to generate sufficient credentials to access resources in different domains. Security information is expressed in the form of assertions about subjects, where a subject is an entity (either human or computer) that has an identity in some security domain.

The framework is made up of SAML Assertions and SAML Protocols. An assertion is a package of information that supplies one or more statements made by an issuer. SAML assertion statements include authentication, authorisation decisions and attributes. Additional types of assertions may be introduced into SAML assertions by using specific XML elements. The SAML protocol defines a simple request-response protocol for discovering information about SAML assertions held by some authority. The SAML protocol may be bound to a variety of data transfer protocols.

SAML provides a format for describing authentication, authorisation and other information that may impact on policy-based decisions.

XACML

Extensible access control mark-up language (XACML) is an XML-based framework for expressing policies in respect of information access over the Web (Klyne, 2002). XACML is intended to address fine-grained access control of authorised activities. It also suggests a policy authorisation model for the implementation of authorisation mechanisms.

The main components of this framework are rules, policies, policy sets, decision requests and XACML context.

- A *rule* is a simple expression that can be evaluated based on the available information.
- A *policy* is a set of rules together with a specified procedure for combining the results of their evaluation.
- *Decision requests* are requests for authorisation decisions.
- *XACML context* is a common abstraction for mapping of policy decision requests.

XACML tries to address areas of policy-based decision making concerned with access control of resources.

2.4.3. Concluding remarks

This section presented a discussion of existing security frameworks from subject literature. The security frameworks discussed were categorised according to the context of their applications. These categories included general Web applications, Web services, the Semantic Web, and XML-based security standards. The discussion of existing security frameworks indicates the current state of affairs in terms of efforts that are being made to provide security in the Semantic Web and the current trend in development of security frameworks. Table 2.3 below lists the existing security frameworks and their application categories.

Table 2.3: Existing security frameworks

APPLICATION CATEGORY	SECURITY FRAMEWORK
General web applications	Digital distributed systems security architecture
	KAoS: Policy and domain services framework
	Security architecture for mobile intelligent agent systems
	Security framework for peer-to-peer grid computing
	Policy-based security framework for Web applications
	Security architecture for open collaborative environment
	Security framework for distributed brokering systems
Web services	Policy-based security framework for business process
	Security framework for Web services
	Using semantic rules to determine access control
	Security for DAML Web services
	Me-Services: Framework for secure and personalised services
	Semantic-based user privacy, protection framework
Semantic web	Policy-based security approach for the Semantic Web
	Concept-level access control for the Semantic Web
	Semantic Web security infrastructure
	Policy-based security model for the Semantic Web
	RDF for policy specification and enforcement
XML-based security standards	XMLDSig
	XMLEnc
	XKMS
	XACML
	SAML
	WS-Security

2.5 CONCLUSION

This chapter provided an overview discussion of the Semantic Web. A detailed discussion of the technologies and functionalities of the Semantic Web architecture was provided in Section 2.2. Technologies adopted by the W3C include *Unicode, URI, XML, RDF, RDF-S, OWL, XMLDSig, and XMLEnc*.

The discussion of the security aspects for the Semantic Web discussed in Section 2.3 included protected entities, security threats, and security services. The protected entities discussed are *agents, Web services,* and *Web resources*. Security threats to Semantic Web entities include *eavesdropping, wire-tapping, session hijacking, impersonation, spoofing, phishing, illegal inference, masquerading, denial-of-service*, *information modification, deletion of resources* and *cloning*. The desired security services for the Semantic Web are *authentication, authorisation, integrity, confidentiality, non-repudiation,* and *availability*. The discussion of existing security frameworks measures the existing efforts to develop security frameworks for future World Wide Web computing.

Section 2.4 provided a brief discussion of existing security frameworks. The discussion of existing security frameworks will assist in establishing their applicability to the Semantic Web. From the definition of a framework ‘a brief set of ideas for organising a thought process about a particular type of thing’ provided in Section 2.4.1, security frameworks discussed included infrastructures, architectures, and models.

CHAPTER 3: RESEARCH DESIGN AND METHODOLOGY

3.1. INTRODUCTION

This chapter presents the research design and methodologies followed in this study. Furthermore, the process of how to reach the intended research results is provided. The purpose of the study is to compile a security framework for the Semantic Web. To achieve the purpose of this study, five objectives were identified in Section 1.3. This chapter provides an outline of the research design and methods followed in reaching the objectives of the study.

In Section 3.2 the research design, the research approach, as well as the strength and weaknesses of the approach are discussed. Furthermore, the theory behind the selected design is also provided. Section 3.3 provides an overview discussion of the selected research methods, including the aims and the justification for using the selected methods. Section 3.4 gives a detailed discussion of the research methodology, which includes data sources and collection methods, and analysis techniques. Section 3.5 concludes the chapter by giving a summary of the research design and the structure of the dissertation in relation to research methods.

3.2. RESEARCH DESIGN

A research design is defined as a plan of how one intends to conduct research (Mouton, 2005:55). A research design focuses on the end product of the research process, that is, the type of study being planned and the type of results aimed at. Its point of departure is the research problem, and hence it focuses on the type of evidence required to address the problem.

The discussion of research design by Mouton (2005:144) identified several dimensions into which a research design can be classified. The first dimension, which is relevant to this section, is that of *empirical* versus *non-empirical* studies. Empirical studies involve observing and measuring reality, thereby confirming knowledge through direct experience. Non-empirical (theoretical) studies involve developing and exploring theories that account for given data.

The second dimension is that of the nature of data used in the study. Data used in empirical studies can be numeric, textual or a combination of both. When the basic data used in an empirical study consist of words, the research is classified as *qualitative*, whereas if the data used are numeric, the research is classified as *quantitative*. A research design may also combine quantitative and qualitative methods to achieve a more rounded and reliable result than either method can give in isolation. The following section presents the classification of this study based on the two dimensions outlined above.

3.2.1. Design classification

In formulating the research problem as research questions (Mouton, 2005:53), a distinction should be made between empirical and non-empirical questions. From the research questions identified in Section 1.2, we categorise those questions that address real-life problems (world 1) as empirical questions and questions of theoretical linkage (world 2) or conceptual models as non-empirical questions. According to Mouton (2005:137), world 1 context involves the world of everyday life and lay knowledge (non-scientific knowledge). In other words, world 1 involves the ordinary social and physical reality that we exist in. World 2, which is the world of science and scientific research, involves the search for ’truthful knowledge‘, i.e., to generate valid and reliable descriptions, models and theories of the world (Mouton, 2005:138).

This study involves the following non-empirical research questions:

1. What security aspects are related to the Semantic Web?
2. What are the requirements of a security framework for the Semantic Web?
3. What are the components that we can use from existing security frameworks?

4. What are the components of a security framework for the Semantic Web?

In order to answer these questions, different non-empirical methods were followed. A *literature review*, which included document analysis, was used to answer sub-questions one, two and three. A *model building* approach was suggested to answer sub-question four. In addition, the study uses *application scenarios* as a proof-of-concept. The use of application scenarios is a qualitative empirical study to strengthen the research validity. All these research methods make use of textual data, and therefore the study is classified as a *qualitative* study. Table 3.1 below summarises the classification of this study in terms of the dimensions discussed above.

Table 3.1: Classification of the research design

Research Method	**Dimension 1**	**Dimension 2**
Literature review	Non-empirical	Qualitative
Model building	Non-empirical	Qualitative
Application scenarios	Empirical	Qualitative

3.2.2. Qualitative studies

Qualitative studies are research approaches in which the basic data used in the research process consist of words or languages (Olivier, 2004:111; Mouton, 2005:53). Qualitative study is useful where an in-depth understanding of a particular situation is required. Since the qualitative research does not involve numerical data, it is not amenable to direct measurement, and therefore the researcher must convince others that the research is reliable.

In qualitative research the reliability and validity of the research are assessed in terms of *auditability*, *credibility* and *comprehensiveness*. Auditability involves the repeatability of the research process. Credibility looks at whether the research results are internally valid, for example: Is the explanation given for the results the only valid explanation? The comprehensiveness aspect ensures an in-depth description of subjects and their relationship to their context.

Another concept related to qualitative studies is that of the researcher's philosophy. Mouton (2005:141) includes positivism, interpretivism and critical theory as the epistemological research approaches. In *positivism*, a positivist believes that there is such a thing as absolute truth and therefore an answer exists that is the truth. In this philosophy, the researcher's role is to find the true answer and describe it. In *interpretivism*, an interpretivist feels that reality is too complex to control every variable. The researcher's role, therefore, is to find a coherent way of understanding a situation within a particular context. In *critical theory*, a critical researcher assumes that social reality is built by people historically. The approach followed in this research is *interpretivist* in nature.

The qualitative approach was considered for this research for the following reasons:

1) In order to generate new theories it is necessary to use qualitative methods, as quantitative methods can only be used to make measurements about existing theories and not to provide tools for discovering new theories.
2) In order to be able to concentrate on details in a specific context rather than to focus on generalisation of broad range of contexts, it is necessary to use qualitative methods.
3) Qualitative methods help in reaching a deep, detailed understanding of situations.

3.2.3. Execution order of the studies

The research started with a literature review to answer the first three research sub-questions. The literature review is presented in Chapter 2 of this document as a *theoretical framework*. A model-building study follows to answer the research sub-question 4. The model-building study depends on the findings from the literature review. The model-building study is presented in Chapter 5 as *research findings*. Application scenarios will then follow to strengthen the validity and reliability of the research. Application scenarios depend on the results of both the literature review and the model-building study. The use of application scenarios will be presented in Chapter 6 as *research analysis*. Figure 3.1 below illustrates the order in which the aforesaid studies will be executed.

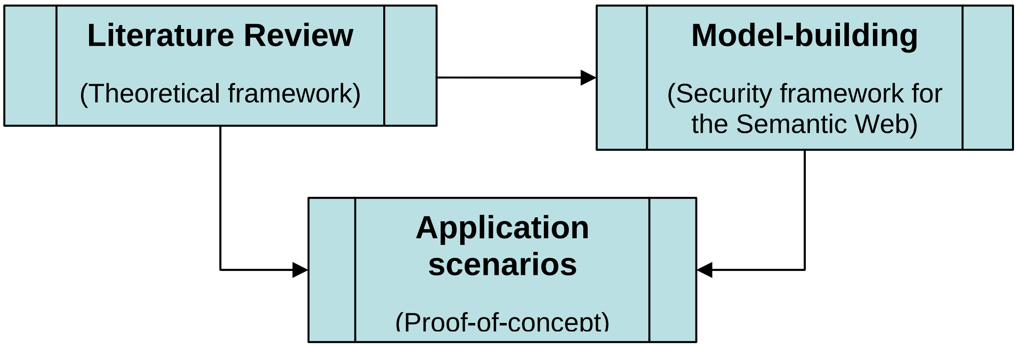


Figure 3.1: Execution order of studies

3.3. RESEARCH METHODS

Different research methods and tools such as surveys, case studies, literature reviews, model-building studies, prototypes, etc. can be used in the field of Information Technology (Olivier, 2004:7; Mouton, 2005:143; Hofstee, 2006: 120). The following sections present a detailed discussion of the selected methods.

3.3.1. Literature review

According to Mouton (2004:179), literature review is defined as a study that provides an overview of scholarship in a certain field through an analysis of trends and debates. A literature review creates a coherent picture of how different concepts fit together. It helps to identify trends in research activity and to define areas of theoretical and empirical weakness (Mouton, 2005:87; Hofstee, 2006:121). A literature review is a non-empirical study (Mouton, 2005; Olivier, 2004), in which the unit of analysis is based on data from an existing academic body of knowledge. Literature reviews rely exclusively on the secondary literature, hence the use of inductive reasoning from a sample of text read to derive a proper understanding of a specific domain of scholarship.

The purpose of conducting a literature review in this study is to provide a sound understanding of the issues and debates in the realm of Semantic Web security. It also provides current thinking and definitions of the Semantic Web terminologies, as well as studying previous works on security frameworks. The literature review in this study involves *document analysis* in which information is extracted from existing literature. The document analysis provides a means to answer the first three research sub-questions and thus to reach the first three objectives of the study. In other words, the document analysis helps the study to describe security aspects related to the Semantic Web, to establish the requirements of a security framework for the Semantic Web and to determine components from existing security frameworks that can be used for the Semantic Web.

The main sources of errors in literature reviews include selection and coverage of sources, selective interpretations of sources, researcher's bias, poor organisation and integration of review, and time factors (Mouton, 2005:90; Hofstee, 2006:121). Although a literature review can produce new theoretical insights, it is limited in producing new, or validating existing, empirical insights (Mouton, 2005:180). The research, therefore, still needs to undertake an empirical study to test the new insights.

The selection of sources for a literature review is based on theoretical factors such as the objectives of the study, research questions, time-frames, etc. The findings from the literature review are presented in Chapter 2, and the extraction of other information from the literature is presented in Chapter 4.

3.3.2. Model-building study

Model-building studies are defined by Mouton (2005:176) as studies that aim at developing new models and theories to explain particular phenomena. Model-building studies are used to answer questions of theoretical linkages and coherence between conceptual models. A model can be defined as a blueprint of a system or process that represents particular phenomena in a clear and concise manner (Olivier, 2004:45; Mouton, 2005:176). According to Olivier (2004:45), 'a model captures the essential aspects of a system or process, while it ignores the nonessential aspects'. A model may describe the system in terms of its components, roles and interfaces in the system. An essential model depicts only the essence of the system, neglecting how the system will be physically implemented (Whitten, Entley & Barlow, 1994). Essential models are similar to logic levels of abstraction. An implementation model shows what the system does and how the system is physically implemented (Whitten et al., 1994). It includes the technology to be used to implement the system. Implementation models are similar to physical levels of abstraction. According to Avison and Fitzgerald (2006), a model is an abstraction representing parts of the real world.

A model provides causal accounts of the world and allows one to make predictive claims. By using models, one can bring conceptual coherence to a particular phenomenon and simplify the understanding of our world (Mouton, 2005:177). Models provide simplicity, comprehensiveness, generality, exactness, and clarity in problem-solving researches (Olivier, 2004:49).

Model-building studies are non-empirical studies that utilise secondary data from an existing academic body of knowledge. Olivier (2004:46) identified three objectives of model-building studies, namely *clarification*, *differentiation* and *generalisation*. Tentative models are used to clarify whether the problem does actually exist. Differentiated models make explicit assumptions to address specific forms of the problem in detail. General models cater for most of the different assumptions made in previous models.

Model-building studies are mainly done through either *inductive* or *deductive* reasoning. Deductive reasoning is more formal in that a set of axioms is formulated and used to deduce additional theoretical propositions. Inductive reasoning is commonly used in statistical model-building, where a model is built to explain particular empirical data. For non-empirical qualitative research, *analogical reasoning*, which is a variation of inductive reasoning, is used. In analogical reasoning a model of a phenomenon is constructed on the basis of its similarities to another phenomenon (Mouton, 2005:177).

The purpose of using model-building study in this research is to develop a security framework for the Semantic Web. The model-building study helps the researcher to answer the fourth research sub-question: What are the components of a security framework for the Semantic Web? In doing so, the last two objectives of the research will be achieved, namely, to establish components of a security framework for the Semantic Web, and to compile a security framework for the Semantic Web.

Assumptions that are made in specifying a model are the main source of error in model-building studies. Models are limited in that they can make claims that are conceptually incoherent, inconsistent and confusing (Mouton, 2005:177).

The approach that will be followed in compiling the security framework for the Semantic Web will be based on software engineering practices. According to Pressman and Ince (2000), the process of CBSE (Component-Based Software Engineering) involves four steps, namely: (i) the selection of potential components for reuse, (ii) qualifying the components, (iii) adaptation of components, and (iv) integration of the components to the proposed system. The construction of the proposed security framework will therefore involve identifying essential components of a security framework, adapting the essential components to the requirements and lastly, integrating the adapted components to form the proposed security framework for the Semantic Web.

The findings of the model-building study will be presented in Chapter 5 of this dissertation. In the next section a discussion of framework application scenarios is presented as a proof-of-concept.

3.3.3. Application scenarios

The study aims to use application scenarios to provide an in-depth description of a limited number of case scenarios. Application scenarios allow specific cases to be studied in more detail without having an implementation of a particular model. Application scenarios are used in this dissertation because this study does not provide an implementation of the proposed security framework.

Application scenarios are empirical in nature and may use qualitative or quantitative information. The study sets out precisely what is to be studied and how the study is to be performed. It spells out what is expected to be learned from the each case scenario. It also lists the aspects of each case that should be observed. Two techniques are usually used for case selection, namely *literal replication* and *theoretical replication*. In literal replication, cases are selected in such a way that they will test the theory in extreme cases. In theoretical replication, cases are selected in such a way that the theory applies in some cases, and does not apply to other cases (Olivier, 2004:101).

The weakness of application scenarios is similar to case study in that it lacks the generalisation of results and non-standardisation of measurements (Mouton, 2005:150). To obtain more general results, multiple-case scenarios are used. The potential bias of the researcher, especially in case selection, may lead to results of little value (Mouton, 2005:150). The advantage of application scenarios is that they may combine both qualitative and quantitative data in one case (Olivier, 2004:100).

The purpose of using application scenarios in this research is to apply different usage scenarios to the proposed security framework as a proof-of-concept. The intention is to ‘prove’ the concept. In this context the term ‘prove’ refers to demonstrating that the proposed security framework for the Semantic Web works. The proof-of-concept scenarios are used to strengthen the validity of the research results. Different usage scenarios will be selected from the literature and applied to the proposed security framework to show that the proposed framework works. The design and implementation of the application scenarios will be presented in Chapter 6 of this document.

3.4. RESEARCH METHODOLOGY

This section presents a research methodology (research process) to be followed in the execution of the research project. A research methodology focuses on the research process and the kind of tools and procedures to be used in the research project (Mouton, 2005:56). Its point of departure is specific tasks at hand such as case selection, data documentation, etc.; hence the focus on individual steps of the research process.

The approach used in this research is adapted from the approach described by Mouton (2005:99-110) to fit the qualitative study followed. The approach will include identification and selection of data sources, collection or gathering of data, data documentation, data capturing and editing, and data analysis and interpretation. Each of these stages of research process is discussed below.

3.4.1. Identification and selection of data sources

The research makes use of documentary sources, which involves existing textual documents available in electronic and printed media. Secondary data are used as explained on Section 3.2.2 of the research design above.

The data sources used in this research include databases available in the University of South Africa’s (UNISA’s) online library catalogue, articles published in journals available at the UNISA library, applicable textbooks, and the Internet. Most of the documents used were obtained from peer-reviewed journals published by the ACM, IEEE, Springer-Verlag, and Elsevier.

3.4.2. Data collection

In this research project, *textual analysis* is used as a means of data collection. Textual analysis involves both content analysis and textual interpretations. Based on the research departure points, namely the Semantic Web and security frameworks, the contents of the referenced publications were analysed to find their applicability to the study. Furthermore, textual interpretation of a relevant publication led to the identification of additional publications relevant to the study.

3.4.3. Data documentation

Data documentation involves the presentation of data in a clear, complete and unbiased manner to enable conclusions to be drawn. In a qualitative study, textual data can be summarised in tables, figures, and matrices. For instance, the theoretical framework presented in Chapter 2 includes a table with a summary of functionalities and technologies of the Semantic Web architecture layers. In model-building study, tools such as unified modelling language (UML) will be used to document the proposed model in terms of its components and interactions between different Web entities.

3.4.4. Data capturing and editing

As noted by Mouton (2005:108), textual data are rich in meaning and difficult to capture in a short and structured manner. Relevant publications were summarised to capture the information provided by the reference. Similar information from different publications were categorised and grouped to simplify and obtain a coherent understanding of a particular domain. The theoretical framework was then developed and presented in Chapter 2 of this document.

3.4.5. Data analysis and interpretation (synthesis)

According to Mouton (2005:108), 'data analysis involves breaking up the data into manageable themes, patterns, trends, and relationships'. In order to understand the different constitutive elements of the data, an inspection is carried out on the relationships between concepts. The aim of the data analysis stage is to turn the data into the evidence for the research findings. For the purpose of this dissertation, data analysis will involve *theoretical findings* such as presentation of the security framework for the Semantic Web coupled with *descriptive findings* such as identification of interesting and significant patterns in existing security frameworks.

Data interpretation involves the deduction of data into larger coherent wholes (Mouton, 2005:108). For the purpose of this study, the interpretation of data involves relating the research results to the existing theoretical framework presented in Chapter 2. The aim of the data interpretation stage is to show whether the existing theoretical framework is supported or falsified by the research findings. In this dissertation, interpretation of data will include application of different usage scenarios to the proposed security framework for the Semantic Web. The research findings and the synthesis will be presented in Chapter 4, Chapter 5 and Chapter 6.

3.5. CONCLUSION

This chapter presented the research design and methodology to be followed in execution of the research process. The research follows a *qualitative* approach based on secondary, textual data. The approach uses both *non-empirical* and *empirical* methods to reach the research objectives. Two non-empirical methods, namely *literature review* and *model-building study,* have been identified for answering the non-empirical research questions. One empirical method, namely *application scenarios* will be used to strengthen the validity and reliability of the research results.

The literature review is presented in Chapter 2 and Chapter 4, the model-building study is presented in Chapter 5 and application scenarios are presented in Chapter 6. Table 3.2 below summarises the relationship between the research questions, research methods and the dissertation chapters.

Table 3.2: Organisation of studies on the dissertation

Research question	**Research method**	**Dissertation chapter**
What security aspects are related to the Semantic Web?	Literature review (document analysis)	Chapter 2
What are the components that we can use from existing security frameworks?	Literature review	Chapter 4
What are the requirements of a security framework for the Semantic Web?	Literature review (document analysis)	Chapter 4
What are the components of a security framework for the Semantic Web?	Model-building study	Chapter 5
	Application scenarios	Chapter 6

CHAPTER 4: ANALYSIS OF EXISTING SECURITY FRAMEWORKS

4.1. INTRODUCTION

This chapter presents the analysis of existing security frameworks discussed in Section 2.4. The analysis of existing security frameworks aims at determining the applicability of existing security frameworks to the Semantic Web context. The analysis will help in establishing the essential components of a security framework and the requirements of a security framework for the Semantic Web. Evaluation of existing security frameworks against the requirements of a security framework for the Semantic Web will help in determining the security frameworks that may be adapted to satisfy the requirements.

From the definition of a framework in Section 2.4.1 i.e. a set of ideas for organising a thought process about a particular type of thing, the existing security frameworks will also be analysed to show their conformity to the definition of a framework.

In the interests of brevity, the following abbreviations presented in Table 4.1 will be used to represent the frameworks in the analysis of existing security frameworks.

Table 4.1: Abbreviations used for existing security frameworks

Abbreviation	Security framework
SWSI	Semantic Web Security Infrastructure
PBSASW	Policy-Based Security Approach for the Semantic Web
PBSMSW	Policy-Based Security Model for the Semantic Web
CLACSW	Concept-Level Access Control for the Semantic Web
PBSFWA	Policy-Based Security Framework for Web Applications
PBSFBP	Policy-Based Security Framework for Business Process
SFDBS	Security Framework for Distributed Brokering Systems
DDSSA	Digital Distributed Systems Security Architecture
SFWS	Security Framework for Web Services
USRAC	Using Semantic Rules to determine Access Control
SDWS	Security for DAML Web Services
SFPPGC	Security Framework for Peer-to-Peer Grid Computing
SAMIAS	Security Architecture for Mobile Intelligent Agent Systems
SAOCE	Security Architecture for Open Collaborative Environment
Me-Services	Framework for Secure and Personalised Services
KAoS	Policy and Domain Services Framework
SBUPPF	Semantic-Based User Privacy, Protection Framework
RDF-PSE	RDF for Policy Specification and Enforcement

Section 4.2 provides general observations from the existing security frameworks. Section 4.3 outlines aspects that are common to the majority of existing security frameworks. Section 4.4 extracts the requirements of a security framework for the Semantic Web. Section 4.5 evaluates the existing security frameworks against the requirements of a security framework for the Semantic Web. Section 4.6 extracts essential components of a security framework. Section 4.7 discusses the adopted security frameworks for adaptation purposes. Section 4.8 concludes the chapter by summarising the research findings from the chapter.

4.2. GENERAL OBSERVATIONS

The surveyed literature indicates that there are relatively security frameworks that were designed specifically for the ***Semantic Web***. The security frameworks that were designed specifically for the Semantic Web include SWSI, PBSASW, PBSMSW, CLACSW, and RDF-PSE. Other security frameworks discussed were designed to provide security in ***Web Services.*** Such frameworks include Me-Services, KAoS, SFWS, SDWS, SBUPPF, PBSFBP, and USRAC. There are several security frameworks in the literature that are generic for ***distributed systems*, including DDSSA, SAMIAS, SAOCE, SFDBS, SFPPGC and PBSFWA.

Most of the frameworks discussed are ***policy-based*** e.g. SWSI, PBSASW, PBSMSW, PBSFWA, PBSFBP, DDSSA, SDWS, SAOCE, KAoS, Me-Services, SBUPPF, and RDF-PSE. These frameworks use security policies to provide different security services. Policy-based approaches are becoming popular in dynamic system adjustability as they have benefits in reusability, efficiency, automation and reasoning about systems behaviour (Uszok, Bradshaw, Johnson, Jeffers, Tate, Dalton, & Aitken, 2004b).

Frameworks such as SWSI, PBSASW, PBSMSW, CLACSW, PBSFBP, SDWS, USRAC, SAMIAS, SAOCE, Me-services, KAoS, SBUPPF, and RDF-PSE provide ***security at semantic level***. These frameworks make use of RDF, RDF-S, DAML+OIL, or OWL to describe security requirements and capabilities of entities.

Frameworks that provide security to the ***derived data*** i.e., that prevent illegal inference, are CLACSW and USRAC. CLACSW uses inferable relationships of concepts to control inferences to unauthorised data, whereas USRAC uses a document filtering mechanism to prune unauthorised information.

Frameworks that incorporate ***conventional security*** solutions include SFWS, PBSASW, SFPPGC, DDSSA, SWSI and SDWS. The conventional security solutions involved include PKI, SSL, X.509, Kerberos, SSH, open-PGP, SPKI, and XKMS.

Most of the XML-based security standards focus on protecting the transmission of documents. For instance, XMLEnc supports end-to-end encryption of XML objects to provide end-to-end network security. The XML-based security standards can be incorporated into a security framework to provide specific security functionality. For example, a security framework may need to utilise XKMS for key distribution and management.

Each of these frameworks contributes in one way or another to understanding the contents of a security framework and the different approaches that can be used to develop a security framework. For the current problem of the Semantic Web, the focus will be on those frameworks that are policy-based, provide security at semantic level, provide security to derived data, and allow the incorporation of conventional security solutions and XML-based security standards.

4.3. COMMON ASPECTS OF ANALYSED SECURITY FRAMEWORKS

The analysis of the existing security frameworks identified several common aspects of security frameworks. This section presents some of those aspects that are of interest to this chapter.

4.3.1. Policy-based security frameworks

The majority of existing security frameworks discussed in Section 2.4 use a policy-based approach to provide security in open distributed environments. A policy is a means of defining rules and constraints applicable to the operation of entities. A security policy is a statement of the security aspects that a system is expected to enforce. In Web applications, policies are used to constrain and regulate a system's behaviour dynamically without requiring the cooperation of the components being regulated.

Benefits of policy-based approaches include reusability, efficiency, extensibility, context sensitivity, verifiability, support for both simple and sophisticated components, protection from malicious components, and reasoning about their behaviour (Uszok et al., 2004b). Specifying policies makes it possible to define a system that is capable of protecting Web resources. For instance, access control policies can be high-level rules that regulate which entities are allowed to access certain resources in a specific manner.

4.3.2. The use of Semantic Web technologies

Recent trends show an increase in the use of Semantic Web technologies in the design and implementation of security frameworks for open distributed systems. Various security ontologies that can represent security information in an intelligible manner have been developed (Denker et al., 2003; Uszok et al., 2004a). Semantic models such as DAML-S, RDF-S and OWL are used to represent different security aspects of domain knowledge (Fensel, 2000). Domain knowledge refers to the common security upper ontology that can be shared by various entities and includes description of the resources, users, environment, and context (Kagal, Berners-Lee, Connoly & Weitzner, 2006). The use of Semantic Web technologies enables the automation of security processes and provides a dynamic and adaptive environment for intelligent enforcement of security mechanisms.

4.3.3. The incorporation of XML-based security standards

Most of the security frameworks discussed in Section 2.4 allow the incorporation of XML-based security standards to provide specific security services. The incorporation of XML-based security standards is partly due to the fact that most of the frameworks use XML as a standard for policy expression. The advantage of incorporating XML-based security standards is that these standards could be mapped into semantic-based representations (Tan & Poslad, 2004b).

4.3.4. Component-based security frameworks

Normally, Web-based applications are developed by using component-based technologies (Pressman & Ince, 2000; Ventuneac et al., 2003). The majority of security frameworks discussed in Section 2.4 are component-based. These frameworks utilise different components or modules for different functionalities of the framework. A component is a basic unit of a system that is capable of performing a particular function. It is an independent and replaceable part of a system that provides a clear function in a particular context (Pressman & Ince, 2000). The advantage of the component-based approach is the ability to implement the *separation of concerns* principle and to improve the flexibility of a system. In a component-based approach, one could change the implementation of the component without affecting the functionality of the system.

4.4. REQUIREMENTS OF A SECURITY FRAMEWORK FOR THE SEMANTIC WEB

4.4.1. Introduction

As stated in Section 2.3, the security aspects related to the Semantic Web are derived from the fundamental security aspects related to an entity's gaining access to computer network resources or services. Security aspects related to the Semantic Web as discussed in Section 2.3, together with challenges presented by the dynamic, autonomous and open nature of the Semantic Web, are used as the basis for establishing the requirements of a security framework for the Semantic Web.

The requirements of a security framework for the Semantic Web will form a basis for establishing the characteristics, components and functionalities of the security framework for the Semantic Web. In other words, the requirements set out what components the security framework should have, the characteristics of the components established, and the security functionalities that should be provided by the security framework.

4.4.2. Discussion of the requirements

The requirements of a security framework for the Semantic Web are derived from the literature surveyed. The requirements are extracted from existing security frameworks discussed in Section 2.4. Different authors have indicated different aspects that should be considered when developing a security framework for the Semantic Web (Finin & Joshi, 2002; Kagal et al., 2003; Park, 2003; Thuraisingham, 2003; Ashri et al., 2004). These aspects range from technologies to be used, security functionalities to be provided, and design criteria, to implementation issues. Figure 4.1 below illustrates the process used to extract the requirements of a security framework for the Semantic Web.

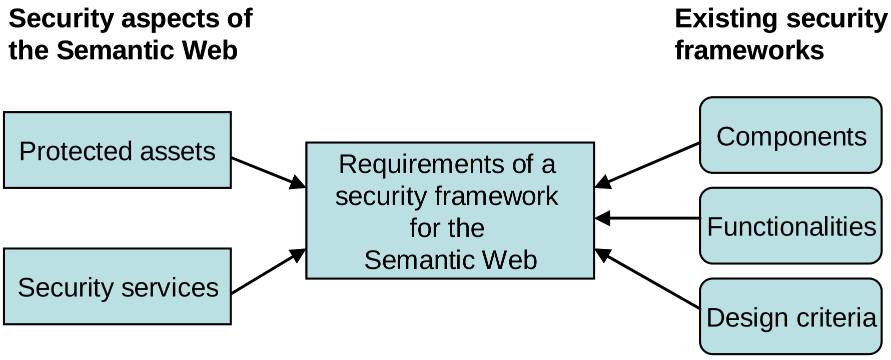

Security aspects of the Semantic Web
Existing security frameworks
Protected assets
Security services
Requirements of a security framework for the Semantic Web
Components
Functionalities
Design criteria

Figure 4.1: Extraction of the requirements

The requirements discussed below are dealt with in no particular order or importance in relation to one another. At this juncture, it is only important that a security framework for the Semantic Web should satisfy all the requirements discussed below.

4.4.2.1. Decoupling of security functionalities from core service functionalities

According to the discussion of infrastructure capabilities for the Semantic Web Security Infrastructure by Ashri et al. (2004), 'the security infrastructure should take into account the possibility that not all services will be able to individually reason about security requirements'.

Since not all Web Services will be able to reason individually about security requirements, such Web Services should be supported by other components that are able to reason about security. The inability of other entities to reason about security requires decoupling of the capability to reason about security from the core service capability.

The advantage of decoupling of security functionality from core service functionalities lies in the increased efficiency of the Web service and the reusability of the security components.

Therefore, the first requirement of a security framework for the Semantic Web will be *decoupling of security functionality from core service functionality.*

4.4.2.2. Layered security support

According to Ashri et al. (2004), 'If an individual Web Service defines and is able to reason about its own security requirements, these may still need to be aligned with the security requirements of the larger domain within which the service operates'. This argument carries with it the need for a layered security framework with security requirements that are Web Service-specific, application-specific, domain-specific and inter-domain-specific. For instance, an application-specific security policy should be aligned to domain-specific security requirements, which should be aligned to generic security requirements (Park, 2003).

To facilitate collaboration, groups of software components, people, resources, and other entities are structured into organisations of domains and sub-domains (Uszok et al., 2004b). Securing entities in these multi-domain environments needs generic security functionalities as well as application-specific security functionalities.

A layered security framework will enable interoperability of Web services between organisations without compromising the security of the service providers.

Therefore, the second requirement of a security framework for the Semantic Web will be *layered security support.*

4.4.2.3. Flexible, dynamic and adaptive

The set of entities that need to access an information source or interact with a given Semantic Web entity can not be enumerated a priori (Finin & Joshi, 2002; Kagal et al., 2003; Thuraisingham, 2003). Entities can also join or leave the Semantic Web without prior notification. The framework must be flexible enough to be applicable in different scenarios with few or no changes (Yague, Mana, Lopez & Troya 2003).

The large number of unknown entities requesting access to the Semantic Web resources calls for the security framework to provide security without necessarily authenticating the requester completely. These behaviours necessitate the Semantic Web security framework to be highly flexible. The framework should also provide flexibility in specifying security requirements and should give every entity a certain amount of autonomy in making its own security decisions (Kagal et al., 2003).

Interactions within the Semantic Web entities can be secured depending on aspects such as the current context, interaction type, and so on. The security framework should be able to adapt to these aspects in order to allow the sending and receiving of appropriate messages (Ashri et al., 2004). The framework must be capable of adapting itself to frequent changes in parameters such as access criteria, client attributes, environment conditions, resources available, and the like (Yague et al., 2003).

Policy-based systems should permit not only new policies to be specified dynamically on demand as new situations occur, but should also allow existing policies to be adapted to meet new changes (Toninelli, Montanari, Kagal & Lassila, 2006).

Therefore the third requirement of a security framework for the Semantic Web is that it be *flexible, dynamic and adaptive.*

4.4.2.4. Semantically rich

The problem of semantic meaning of the security information where it is not feasible to expect all entities to use the same terminology to represent security protocols and information necessitates the security framework for the Semantic Web to be semantically rich (Finin & Joshi, 2002; Kagal et al., 2003).

There is an increasing need to be able to describe and reason about security requirements at the semantic level (Ashri et al., 2004, Thuraisingham, 2003). Security should be preserved at the semantic level in order to provide access control at the finest granularity and to ensure that RDF documents are secured (Thuraisingham, 2003).

A semantically rich representation allows description of contexts at a high level of abstraction, which is essential in both reasoning and conflict resolution for policies (Toninelli et al., 2006).

It is important for the Semantic Web entities to be able to express their security requirements clearly and concisely to avoid ambiguous interpretation of security information. The utilisation of the Semantic Web by software agents requires the annotation of semantics of data within the Web. To secure the Semantic Web therefore needs security descriptions at the semantic level.

Therefore, the fourth requirement of a security framework for the Semantic Web is that it be *semantically rich.*

4.4.2.5. Simple enough to automate

The need for machines to access and process information securely on the Semantic Web requires the automation of the Semantic Web security framework (Finin & Joshi, 2002).

According to Kagal et al. (2003), the ability to handle security and privacy and the ability to automate security protocols for the use of all Web entities are the key needs for the vision of the Semantic Web to succeed. Complex and static security mechanisms that will need the intervention of system administrators will not scale well with the Semantic Web.

Too sophisticated security mechanisms to implement security requirements in open environments can result in complex systems that are impractical for large-scale interoperable deployments (Tan & Poslad, 2004b).

The move towards more autonomous systems, where decisions are made without direct human intervention, and more complex operating scenarios calls for the automation of the security services (Ashri et al., 2004).

Automatic computing is necessary for systems that need to interoperate securely in open environments, where real-time applications need automatic security mechanisms to interoperate, mediate and self-manage (Tan & Poslad, 2004b).

Therefore the fifth requirement of a security framework for the Semantic Web is that it be *simple enough to automate*.

4.4.2.6. Impervious to common network problems

The Semantic Web as the extension of the current Web shares the inherent common characteristics and problems of the current Web technologies such as network partitioning (Finin & Joshi, 2002). Issues of secure communication channels, user and server authentication, and end-to-end network security are common in network environments.

Security from the above-mentioned common network problems is not specified on the Semantic Web but needs to be addressed by the security framework for the Semantic Web. The security framework for the Semantic Web therefore needs to include mechanisms such as SSL, X.509, etc. that will address common security problems on networks.

The use of conventional security technologies such as PKI for user authentication or SSL for secure communication channels within the

Semantic Web context will support the semantic-based security framework in a holistic security approach (Ashri et al., 2004).

Therefore the sixth requirement of a security framework for the Semantic Web is that it should be *impervious to common network problems.*

4.4.2.7. Implementable on the current Semantic Web technologies

The Semantic Web is built upon layers of expressive languages of increasing powers. These languages enable the automation of the retrieval and usage of Web resources. Languages that have been adopted by W3C include Unicode/URI, XML, RDF, and OWL (Berners-Lee, 2000; Berners-Lee et al., 2001; Horrocks et al., 2005).

Since the study focuses on compiling the security framework for the Semantic Web and not on developing or improving the current enabling technologies, the security framework for the Semantic Web needs to be implementable on existing technologies.

Therefore the seventh requirement of a security framework for the Semantic Web is for it to be *implementable on the current Semantic Web technologies.*

4.4.2.8. Provides protection to all Semantic Web entities

In order to enforce security on a computer system, one needs to identify computing assets that need to be protected (Pfleeger & Pfleeger, 2003).

In the context of the Semantic Web, the assets that need protection are the entities that interact with the Semantic Web. These entities, which were discussed in Section 2.3.2, include agents, Web services and Web resources.

Web services need to be protected from session-hijacking, eavesdropping, wire-tapping, impersonation, spoofing, masquerading, and denial-of-services.

Web resources need to be protected from eavesdropping, wire-tapping, impersonation, deletion of resources, illegal inferences and denial of services.

Software agents need to be protected from eavesdropping, information modification, masquerading, cloning and denial of service.

Therefore, the eighth requirement of a security framework for the Semantic Web will be to *provide protection to all entities that interact with the Semantic Web*.

4.4.2.9. Provides a complete set of security services

According to a computer security principle, i.e., the *principle of easiest penetration*, an intruder must be expected to use any available means of penetration. This principle teaches us that to attain the goal of security one needs to consider every possible means of protection (Pfleeger & Pfleeger, 2003). For instance, if the authorisation mechanisms are strong but the authentication mechanisms are weak, intruders may abuse the authentication system to be able to compromise the authorisation system without being detected.

To achieve security, the framework should provide all security services discussed in Section 2.3.4, i.e. authentication, authorisation, integrity, confidentiality, availability and non-repudiation.

Therefore, the ninth requirement of a security framework for the Semantic Web will be to *provide a complete set of security services.*

4.4.3. Concluding remarks

In order to develop a security framework for the Semantic Web, one needs to set criteria for the aspects that need to be included in the proposed framework. These criteria will be used, not only in development of the framework, but also to establish how the existing security frameworks can be used to solve the current problem of the Semantic Web. The requirements discussed above are summarised in Table 4.2 below.

Table 4.2: Requirements of a security framework for the Semantic Web

No	Requirements
1	Decoupling of security functionalities from core service functionalities
2	Layered security support
3	Flexible, dynamic and adaptive
4	Semantically rich
5	Simple enough to automate
6	Impervious to common network problems
7	Implementable on current Semantic Web technologies
8	Provides protection to all Semantic Web entities
9	Provides a complete set of security services

4.5. EVALUATION OF EXISTING SECURITY FRAMEWORKS

4.5.1. Introduction

In this section, the existing security frameworks are evaluated against the requirements of a security framework for the Semantic Web as discussed in Section 4.4 above to determine their applicability to the current study.

4.5.2. Evaluation against the requirements of a security framework for the Semantic Web

In the evaluation of existing security frameworks against the requirements of a security framework for the Semantic Web, each requirement will be compared with existing security frameworks and those frameworks that adhere to the requirement will be identified.

Requirement 1: Decoupling of security functionalities

As discussed in Section 4.5.2, the decoupling of security functionalities from core service functionalities increases the efficiency of Web services and reusability of security components. The various security frameworks discussed indicate the importance of decoupling security functionalities from core service functionalities.

Security frameworks such as SWSI, PBSASW, PBSMSW, PBSFWA, USRAC, SDWS, SAMIAS, and SAOCE incorporate this requirement in their design principles.

By taking into account that not all entities will be able to reason about security requirements, the SWSI enforces the decoupling of security services by introducing a Semantic Firewall to reason about security requirements of entities. The PBSASW uses a policy engine contained in a Web server to interpret and reason about policies. The Web server is trusted to perform security functionalities for Web entities. The SDWS uses the security reasoner to perform reasoning about security requirements and capabilities, whereas the service matchmaker performs other core service functionalities. The PBSMSW separates the security mechanisms and security applications from the reification layer that includes the service description. The separation enables the security conceptual model to be independent of the application requirements. The modular approach used by the PBSFWA to implement security services enables the decoupling of security functionalities.

Other security frameworks such as WS-S, PBSFBP, SFDBS, DDSSA, SFWS, SFPPGC, Me-Services, KAoS, SBUPPF, and RF-PSE do not support the decoupling of security functionalities from core service functionalities.

Requirement 2: Layered security support

A layered security framework enables interoperability of Web services in a multi-agent multi-domain environment (Tan and Poslad, 2004a).

Security frameworks that support layered security include SWSI, PBSASW, PBSMSW, CLACSW, XACML, SDWS, SAMIAS, SAOCE, Me-Services, KAoS, SBUPPF, and RDF-PSE.

The SWSI supports layered security by dictating the infrastructure capabilities to align the service-specific security requirements to organisation-specific security requirements, which may also be aligned to the security requirements of the specific domain. The PBSASW enforces security on agents at two levels, i.e. at agent level and at platform level (agent environment). The SDWS describes security concepts at three levels, namely, capabilities, process and invocation. The SAOCE uses a job-centred approach in which security is defined at two levels, namely, the business part and the technical part. The KAoS security functionalities are categorised into generic and application-specific. The RDF-PSE specifies high-level policies for general domain, and scenario-specific policies conforming to the general domain. SAMIAS supports layered security by organising entities into domains made up of authentication centres and several authentication agents. A domain can be a LAN, an organisation’s VPN or a group of machines.

Security frameworks that do not support layered security include PBSFBP, SFDBS, DDSSA, SFWS, and SFPPGC.

Requirement 3: Flexible, dynamic and adaptive

Security frameworks that are flexible, dynamic and adaptive include SWSI, PBSASW, PBSMSW, PBSFWA, PBSFBP, SFPPGC, SAMIAS, SAOCE, Me-services, KAoS, and RDF-PSE.

The infrastructure capability of the SWSI ensures that the security infrastructure is able to adapt to the current context in order to allow necessary messages. In the PBSASW entities are allowed to delegate rights dynamically. The PBSMSW policy layer uses a reasoning model to support dynamic reconfiguration of security services. The PBSFWA uses a dynamic adaptive authentication policy to specify authentication mechanisms to be used in different authentication contexts. The PBSFBP uses an aspect-oriented approach to specify policies that are implemented at the process level to be enforced dynamically. In the SFPPGC the set of security requirements is determined by peer, and can be changed without recompiling the application. The scalable naming and identification schemes of the SAMIAS support potential dynamic open-ended growth in number and size of agents that need to interact in large distributed systems.

Security frameworks that are not flexible, dynamic or adaptive include CLACSW, SFDBS, DDSSA, SFWS, USRAC, SDWS and SBUPPF.

Requirement 4: Semantically rich

Security frameworks that are semantically rich include SWSI, PBSASW, PBSMSW, CLASW, PBSFBP, USRAC, SDWS, Me-services, KAoS, SBUPPF, and RDF-PSE.

The SWSI uses OWL ontologies to represent security concepts that are reasoned by the semantic firewall. The PBSASW uses a semantic policy language with ontologies that allow policies to be described in RDF-S. The PBSMSW uses the security ontology model to specify security profiles in DAML+OIL or OWL. In CLACSW, concepts are defined in ontologies represented in OWL. The PBSFBP uses an ontology repository with both business ontology and security ontology. Security ontologies are defined in DAML+OIL. In USRAC the knowledge base used to describe the authorisation process is defined in OWL. Rules that are used to define access rights of individuals are represented in SWRL. The OWL-DL reasoning engine is used to evaluate rules. The SDWS defines security ontologies in DAML+OIL that allow the annotation of Web entities with security concepts.

Security frameworks that are not semantically rich include PBSFWA, SFDBS, DDSSA, SFWS, SFPPGC, SAMIAS, and SAOCE.

Requirement 5: Simple enough to automate

Security frameworks that are simple enough to automate include SWSI, PBSASW, CLACSW, SFWS, USRAC, SDWS, SAMIAS, SAOCE, Me-services, KAoS, SBUPPF, and RDF-PSE. These are frameworks that are not complex, and use technologies such as RDF, OWL, etc. that are easily automated.

Security frameworks that are complex and not easy to automate include PBSMSW, and SFPPGC.

Requirement 6: Impervious to common network problems

Security frameworks that are impervious to common security problems include SWSI, PBSASW, PBSMSW, PBSFWA, PBSFBP, SFDBS, DDSSA, SFWS, SDWS, SFPPGC, SAMIAS, SAOCE, and KAoS.

The SWSI incorporates ‘conventional’ security technologies such as PKI and X.509 for authentication and SSL for secure communication channels. The PBSASW allows existing security mechanisms such as PKI, Kerberos, etc. to be integrated into the framework. The SDWS models the commonly present security features such as authentication and communication security to be implemented by Web-based application servers. The SDWS allows the specification of security requirements such as authentication by X.509 or use of SSH protocol. The PBSMSW uses a security mechanism and specifications layer to specify instances of security concepts and services from existing security standards such as X.509 or SSL. The PBSFWA uses existing Web services’ specifications to define enhancements to provide security services to Web services’ end points and data communication between them.

The PBSFBP allows the specification of security standards and mechanisms such as X.509 and PKI. The SFDBS allows specific security mechanisms such as Kerberos, PKI, etc. to be plugged into specific applications. The DDSSA provides a message authentication service by using message hash functions and secure communication channels by using an encrypted logical path. The SFWS uses SSL to provide confidentiality of UDDI data while it is being transferred to and from UDDI nodes.

Security frameworks that do not provide protection to common network problems include CLACSW, USRAC, Me-services, SBUPPF, and RDF-PSE.

Requirement 7: Implementable on current Semantic Web technologies

Security frameworks that are implementable on current Semantic Web technologies include SWSI, PBSASW, PBSMSW, CLACSW, PBSFBP, USRAC, SDWS, SAMIAS, SAOCE, Me-services, KAoS, SBUPPF, and RDF-PSE.

The PBSASW allows policies to be described in RDF-S. The PBSFBP uses security ontologies defined in DAML+OIL. The SDWS defines security ontologies in DAML+OIL that allow the annotation of Web entities with security concepts. The PBSMSW specifies security profiles in DAML+OIL or OWL. The SWSI uses OWL ontologies to represent security concepts. In CLACSW, concepts are defined in ontologies represented in OWL. In USRAC the knowledge base used to describe authorisation process is defined in OWL.

Security frameworks that are not implementable on current Semantic Web technologies include PBSFWA, SFDBS, DDSSA, SFWS, and SFPPGC.

Requirement 8: Provide protection to all Semantic Web entities

Security frameworks that provide protection to *agents* include PBSASW, SDWS, SAMIAS, KAoS, and SBUPPF. These frameworks allow agents to specify policies that a requester must satisfy in order to use its services.

Security frameworks that provide protection to *Web services* include SWSI, PBSASW, PBSMSW, PBSFWA, PBSFBP, SFWS, USRAC, SDWS, SFPPGC, SAOCE, Me-services, KAoS, and SBUPPF. Most of these frameworks make use of policies to specify the security requirements of a Web service for authorised access to the Web service.

Security frameworks that provide protection to *Web resources* include SWSI, PBSASW, PBSMSW, CLACSW, PBSFWA, SFDBS, DDSSA, SFWS, USRAC, SFPPGC, SAMIAS, SAOCE, Me-services, KAoS, SBUPPF, and RDF-PSE. The security protecting Web resources is mostly provided by cryptographic methods such as digital signatures, encryptions, and so on. SSL is commonly used for secure communication of Web resources.

Table 4.3: Entities protected by security frameworks

Security frameworks	Protected entities		
	Agents	**Web services**	**Web resources**
SWSI		✓	✓
PBSASW	✓	✓	✓
PBSMSW	✓	✓	✓
CLACSW			✓
PBSFWA		✓	✓
PBSFBP		✓	
SFDBS			✓
DDSSA			✓
SFWS		✓	✓
USRAC		✓	✓
SDWS	✓	✓	
SFPPGC		✓	✓
SAMIAS	✓		✓
SAOCE		✓	✓
Me-Services	✓	✓	✓
KAoS	✓	✓	✓
SBUPPF	✓	✓	✓
RDF-PSE			✓

The above analysis shows that the security frameworks that provide protection to all Semantic Web entities are PBSASW, PBSMSW, KAoS, Me-services, and SBUPPF.

The PBSASW uses distributed policy management, where each Web entity (agent, Web service, Web resource) is capable of specifying policies for its access. The PBSMSW uses the security ontology to specify the protected entities, threats and safeguards. The protected entities include Web service, agents, and Web resources. The KAoS policy ontology specifies authorisation policies for agents who want to interact with services or resources.

Requirement 9: Provide a complete set of security services

Security frameworks that provide authentication services include PBSASW, PBSMSW, PBSFWA, PBSFBP, SFDBS, DDSSA, SFWS, SDWS, SFPPGC, SAMIAS, SAOCE, Me-Services, KAoS, and RDF-PSE.

The majority of existing security frameworks provide authorisation services. Frameworks that provide authorisation services include PBSASW, CLACSW, PBSFWA, PBSFBP, SFDBS, DDSSA, SFWS, USRAC, SDWS, SFPPGC, SAMIAS, SAOCE, Me-services, KAoS, and RDF-PSE.

Security frameworks that provide integrity services include PBSFWA, PBSFBP, SFDBS, DDSSA, SFWS, SFPPGC, SAMIAS, SAOCE, Me-services, KAoS, and SBUPPF. Digital signatures and hash functions are used to ensure integrity of resources on the Web.

Security frameworks that provide confidentiality services include PBSFWA, PBSMSW, PBSFBP, SFDBS, DDSSA, SFWS, USRAC, SFPPGC, SAMIAS, SAOCE, Me-services, and SBUPPF. Cryptographic solutions such as encryption are mostly used to ensure confidentiality of Web resources and communication information.

Security frameworks that provide availability services include PBSMSW, PBSFWA, SFWS, and DDSSA. Security frameworks that provide non-repudiation services include PBSMSW, SFWS, and SFPPGC.

Table 4.4: Security services provided by frameworks

Security framework	Security services					
	Authentication	Authorisation	Confidentiality	Integrity	Availability	Non-repudiation
SWSI	✓	✓	✓	✓		
PBSASW	✓	✓	✓	✓		
PBSMSW	✓	✓	✓	✓		✓
CLACSW		✓				
PBSFWA	✓	✓	✓	✓	✓	✓
PBSFBP	✓	✓	✓	✓		
SFDBS	✓	✓	✓	✓		
DDSSA	✓	✓	✓	✓	✓	
SFWS	✓	✓	✓	✓	✓	✓
USRAC		✓	✓			
SDWS	✓	✓	✓			
SFPPGC	✓	✓	✓	✓	✓	✓
SAMIAS	✓	✓	✓	✓		
SAOCE	✓	✓	✓	✓		
Me-Services	✓	✓	✓	✓		
KAoS	✓	✓		✓		
SBUPPF			✓	✓		
RDF-PSE	✓	✓				

The above analysis shows that the security frameworks that provide a complete set of security services include PBSFWA, SFWS, and SFPPGC.

The PBSFWA uses a dynamic adaptive authentication policy to specify what authentication mechanisms are used in different identification contexts based on user credentials and associated permissions. It uses access control policy to specify protected entities and their protection mechanisms based on a set of user-role-privilege associations. The access control policy also provides integrity and confidentiality services.

The SFWS uses a UDDI-specified API to provide authentication services. SSL 3.0 provides confidentiality of UDDI data. Digital signatures ensure integrity of data. SFWS uses a trust policy infrastructure implemented in WSDL to specify trust processing to be performed on a service request.

The modularity and reconfigurability features of the SFPPGC enable changes and additions to security services without recompiling the whole application. The framework uses symmetric cryptography for authentication confidentiality and integrity. Message authentication codes and asymmetric cryptography are used for integrity and authenticity. Access policies are used for authorisation.

The evaluation of the existing security framework against the requirements of the security framework for the Semantic Web is summarised in Table 4.5.

Table 4.5: Requirements satisfied by security frameworks

Framework	**Requirements Satisfied**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
SWSI	✓	✓	✓	✓	✓	✓	✓		
PBSASW	✓	✓	✓	✓	✓	✓	✓	✓	
PBSMSW	✓	✓	✓	✓		✓	✓	✓	
CLACSW		✓		✓	✓		✓		
PBSFWA	✓	✓	✓			✓			✓
PBSFBP			✓	✓		✓	✓		
SFDBS						✓			
DDSSA						✓			
SFWS					✓	✓			✓
USRAC	✓	✓		✓	✓		✓		
SDWS	✓	✓		✓	✓	✓	✓		
SFPPGC			✓			✓			✓
SAMIAS	✓	✓	✓		✓	✓	✓		
SAOCE	✓	✓	✓		✓	✓	✓		
Me-Services		✓	✓	✓	✓		✓	✓	
KAoS		✓	✓	✓	✓	✓	✓	✓	
SBUPPF		✓		✓	✓		✓	✓	
RDF-PSE		✓	✓	✓	✓		✓		

4.5.3. Concluding remarks

In this section each security framework was evaluated against the requirements of a security framework for the Semantic Web. There is no security framework that satisfies all the requirements, although all security frameworks satisfy at least one requirement. Few security frameworks satisfy more than five requirements. Those frameworks that do satisfy more than five requirements have the potential to be adapted to a security framework for the Semantic Web.

4.6. ESSENTIAL COMPONENTS OF A SECURITY FRAMEWORK

The analysis of existing security frameworks also identified several essential components common to security frameworks. The essential components form basic components of security frameworks. This section presents essential components of a security framework identified from the analysis of existing security frameworks and the security aspects of the Semantic Web.

4.6.1. Knowledge base

This component involves the description of security-related information that will be used in providing security services to knowledge-based systems (KBS). A KBS is a computerised system that uses knowledge about some domain to arrive at a solution of a problem in that domain (Gonzalez & Dankel, 1993; Lee, Padhyaya, Rao, & Sharman, 2005).

Knowledge is a combination of instincts, ideas, rules and procedures that guide decisions and actions (Alter, 1996). Singh and Salam (2006) defined *knowledge* as ‘information in the context of a specific problem domain upon which action can be advised or taken’. Knowledge is a prerequisite for quality decision-making (Lee et al., 2005). A knowledge base is the repository of facts derived from the security ontology model and the process ontology model. It contains sets of facts and rules (Alter, 1996). Ontology is a shared formal conceptualisation of a particular domain (Decker et al., 2000a). Security ontology models are used to describe security concepts and their relations. Process ontology models[2] are used to specify allowed interaction scenarios or business workflows.

4.6.2. Security services

This component describes security services provided by the framework. The security services are related to the protected Web entities and their security threats. The component provides the method for defining security goals such as authentication, integrity, or confidentiality, and their required input and output parameters for specific Web entities. The security services to be included are determined by the security threats of protected entities as discussed in Section 2.3.3 of this dissertation. Table 4.6 below summarises security services.

[2] http://www.daml.org/services/owl-s/1.0/

Table 4.6: Security services for the Semantic Web

Security service	Description
Authentication	Verifying that an entity is indeed who it say it is
Authorisation	Deciding which subjects should have what type of access to which objects
Availability	Assets are accessible to authorised parties at appropriate times
Confidentiality	Assets are accessed only by authorised parties
Integrity	Assets can be modified only by authorised parties and in authorised ways
Non-repudiation	Clients can not deny having sent a message or performed a transaction

4.6.3. Policies, rules and constraints

This component defines sets of policies, rules and constraints that can be applied to entities that interact with the environment. This component involves security policies, business rules, and other constraints such as speech acts. Ontology is used to represent security constraints as policies to allow flexibility and dynamic representation of several types of constraints (Huang, 2006).

The advantage of semantic policy description is the ability to model non-functional properties into the policies and reasonings about them, and the integration of business rules and knowledge base in specifying policies (Huang, 2006). Policies are used to specify security mechanisms and security services to address specific security threats. Security policies for the Semantic Web should, therefore, address the security threats identified in Section 2.3.3.

Furthermore, policies are used by entities to specify their security requirements and capabilities. Policies are essential components of automatic trust systems (Uszok et al., 2004b). Table 4.7 below summarises security threats to be addressed by security policies.

Table 4.7: Security threats on the Semantic Web

Security threat	Description
Interception	Unauthorised access of computing assets
Interruption	Attacks on computing assets that result in lost, unavailable or unusable assets
Modification	Unauthorised change of computing assets
Fabrication	Creation of counterfeit objects on a computing system

To address security threats to the Semantic Web identified above, security policies such as authentication policy, access control policy, privacy policy, and transaction policy are adopted. The adopted security policies were extracted from existing security frameworks (Ventuneac et al., 2003; Tan & Poslad, 2004b; Turner, Dogac & Toroslu, 2005; Toninelli et al., 2006;).

4.6.4. Security mechanisms

This component describes security standards and specifications that can be utilised by entities to specify their security requirements and capabilities. It involves specific instances of security concepts, policies, and service entities. The security standards to be included are those recommended by W3C and IETF. Conventional security solutions such as PKI, SSL, etc. which are essential in providing security in the case of common network problems such as end-to-end network security are also included. (Ashri et al., 2004).

4.6.5. Reasoning engine

This component involves a reasoning engine or inference engine that invokes a particular security mechanism to provide a specific security service based on the security policy. A reasoning engine uses rules in the knowledge base to decide what to do (Alter, 1996). It is the interpreter of the knowledge stored in the knowledge base. The reasoning engine interprets and reasons about policies and domain information to make decisions about applicable security mechanisms and security services. It also reasons about whether the interacting entities are able to support the required security policies.

Essential components of a security framework are summarised in Table 4.8 below.

Table 4.8: Essential components of a security framework

Component	**Description**
Knowledge base	Stores security-related facts and rules
Security services	Allow specifications of security goals
Policies	Define sets of rules, policies and constraints to address security threats
Security mechanisms	Describe security standards and specifications that provide particular security services
Reasoning engine	Interprets and reasons about policies and security concepts in order to enforce a security service by using a particular security mechanism

4.7. THE ADOPTED SECURITY FRAMEWORKS

4.7.1. Introduction

From the evaluation of existing security frameworks as presented in Section 4.6 above, four security frameworks were selected for adaptation to a security framework for the Semantic Web.

The following criteria were used to select existing security frameworks for adoption. A framework should always –

1) satisfy at least five out of nine requirements as depicted in Table 4.5
2) contain at least three out of five essential components of a security framework as identified in Section 4.6
3) be policy-based for compatibility during integration of components.

The four existing security frameworks that will be used are PBSASW: *Policy-based security approach to the Semantic Web* (Kagal et al., 2003), PBSMSW: *Profile-based security model for the Semantic Web* (Tan & Poslad, 2004b), SAOCE: *Security architecture for open collaborative environment* (Demchenko et al., 2005), and PBSFWA:

Policy-based security framework for Web-enabled applications (Ventuneac et al., 2003). These security frameworks were selected because they satisfy majority of the requirements of a security framework for the Semantic Web and essential components of a security framework. The selected security frameworks satisfy the above criteria as depicted in Table 4.9 below.

Table 4.9: Adopted security frameworks

Framework	**PBSASW**	**PBSMSW**	**SAOCE**	**PBSFWA**
Requirements satisfied	8	7	6	5
Essential components	4	4	3	3
Policy-based	✓	✓	✓	✓

The next section provides a brief description of essential components that will be adapted from the selected existing security frameworks.

4.7.2. Essential components of the adopted security frameworks

The policy-based security approach to the Semantic Web (Kagal et al., 2003) makes use of the *distributed policy management* approach. In distributed policy management every entity is able to define and enforce its own security policies.

The advantage of this approach is that it enhances flexibility and interoperability of security mechanisms. Security for the Semantic Web entities is specified as *Web services security*, *Web resources security* and *agent security*. A *semantic policy language* is used to specify policies and to describe context-dependent security requirements of entities. The *policy engine* interprets and reasons about policies and domain information to make decisions about applicable rights, prohibitions, obligations and dispensations. The policy engine has a set of ontologies to represent security concepts and a *process ontology model* to describe workflow.

The *security mechanisms and specification* layer of the profile-based security model for the Semantic Web (Tan & Poslad, 2004b) allows the incorporation of existing XML-based security standards defined by W3C and IETF. The *security ontology model* provides conceptual means to define properties and relations between security, trust and privacy.

The security architecture for open collaborative environment (Demchenko et al., 2005) includes a communication security layer, a policy expression layer and a security services layer that are required for the security framework for the Semantic Web.

The *communication security* layer defines the network security infrastructure such as SSL, IPSec, VPN, etc. to ensure end-to-end network security. The *policy expression* layer defines set of policies that can be applied to entities that interact with the environment. The *security services* layer defines security services such as authentication, authorisation, etc. for secure operation of the environment.

The policy-based security framework for Web-enabled applications (Ventuneac et al., 2003) uses a modular approach to provide security services by using *security standards and mechanisms* and flexible *security policies*. The security standards and mechanisms define specific instances of security objects such as PKI, X.509, etc. Flexible security policies included are authentication, access control, security administration, and accountability.

4.7.3. Concluding remarks

In this section, four security frameworks were adopted for adaptation to a security framework for the Semantic Web. The criteria set for the selection of the frameworks are based on the evaluation of existing security frameworks presented in Section 4.5. Essential components that will be adapted from the adopted security frameworks have also been identified.

4.8. CONCLUSION

The chapter started by identifying common aspects and essential components of existing security frameworks, and was followed by the extraction of the requirements of a security framework for the Semantic Web.

From the analysis of existing security frameworks, the study discovered that the majority of existing security frameworks are *policy-based*, *component-based*, and *utilise XML-based security standards*. The study also established essential components of a security framework. These identified essential components for a security framework including the identification of *security services*, *security mechanisms*, *enforcement of security mechanisms*, *policies and constraints*, and a *knowledge base*.

From the evaluation of existing security frameworks against the requirements of a security framework for the Semantic Web, it is evident that there is no security framework that meets all the requirements of a security framework for the Semantic Web. Only one security framework i.e. *Policy-based Approach to Security for the Semantic Web,* meets eight of the nine requirements. This observation implies a need to develop or adapt some of these frameworks to a framework that meets all the requirements of a security framework for the Semantic Web.

Four security frameworks have been adopted for the adaptation process. The adopted frameworks are policy-based, satisfy at least five requirements, and contain at least three essential components of a security framework.

CHAPTER 5: THE SECURITY FRAMEWORK FOR THE SEMANTIC WEB

5.1. INTRODUCTION

The approach to be followed in the construction of the proposed security framework for the Semantic Web will be based on the adaptation of existing security frameworks to fit the requirements of a security framework for the Semantic Web as discussed in Section 4.4 of this dissertation.

Firstly, security frameworks that satisfy the majority of the requirements will be identified and used in the adaptation process. In the adaptation process the essential components of a security framework identified in Section 4.6 above will be used to determine the features or elements to be used to satisfy a requirement. Lastly, the selected components will be integrated and presented as an adapted and integrative framework. The adaptation process is presented in Section 5.2. This approach is a variation of inductive reasoning called *analogical reasoning,* in which a model of a phenomenon is constructed based on its similarities to another phenomenon (Mouton, 2005:177).

In Section 2.4.1, a framework was defined as a brief set of ideas for organising a thought process about a particular type of thing. A model describes the reality without dealing with every detail of it. This study proposes a framework and uses models to represent the proposed framework. This study is said to propose a framework because it provides a set of ideas (security aspects of the Semantic Web) for organising a thought process (security structure) for a particular thing (the Semantic Web). Therefore, this study proposes a framework according to the definition of a framework provided above. Both frameworks and models are important in science as they help the user to make sense of the world’s complexity.

The documentation of the proposed security framework will be based on the approach suggested by Bass et al. (2003) for documenting a view of a reference model. A reference model is a division of functionalities together with dataflow between the elements. A view is a representation of a coherent set of elements and the relations between them. For the purpose of the proposed security framework, two types of model descriptions will be used from the template presented by Bass et al. (2003).

Firstly, a *primary presentation* that shows the elements and relationships that populate a view will be used to provide an abstraction of the proposed security framework in Section 5.3.2. An abstraction is the process of stripping a system of its concrete or physical features (Avison & Fitzgerald, 2006). It indicates important aspects of a system at various levels.

Secondly, an *element catalogue* that details the elements and relations depicted in the primary presentation will be used in Section 5.3.3 to provide backup information that explains the contents of the primary presentation. Each component depicted in the primary presentation will be explained in detail under the *elements and their properties* section. Functionalities of the framework will be explained under *relations and their properties* in Section 5.3.4. The interactions of the framework's components will be explained under *element behaviour* in Section 5.3.5.

According to Avison and Fitzgerald (2006), the above-mentioned approach of documenting a model is a logical level abstraction of a system. The *logical level* is a description of the system without any reference to the technology that could be used to implement it. To document a proposed system properly, one needs to document a physical level abstraction as well. The *physical level* is a description of the system that includes the technology of a particular implementation. For the purpose of this study a physical level abstraction will be presented in Section 5.3.7 by using a pictorial or schematic model to capture necessary information for implementing the framework. Section 5.4 will present evaluation of the proposed security framework.

5.2. THE ADAPTATION PROCESS

5.2.1. Introduction

The adaptation process will involve modifying the selected components to fit the requirements of a security framework for the Semantic Web and the integration of components from four existing security frameworks adopted in Section 4.7. Section 4.6 dealt with selection of potential components and their qualifications. This section will deal with adaptation and integration of the components.

In the following section, the adaptation process of the adopted security frameworks is presented. The adaptation is based on the integration of the identified components to satisfy the requirements of a security framework for the Semantic Web.

5.2.2. Adaptation of security frameworks to the requirements of a security framework for the Semantic Web

Requirement 1: Decoupling of security functionalities

Certain Semantic Web entities such as Web resources may not be able to reason about security requirements and capabilities of other entities, hence the need for an independent policy engine that will interpret and reason about security information and policies on behalf of Web entities.

The proposed security framework will use a distributed policy management approach in which every entity will be able to define and enforce its own security policies. However, entities which are not able to enforce their own security policies will utilise the application-specific policy engine to enforce their security policies. A policy engine or security reasoner accepts the requirements and capabilities of interacting entities as input and decides to what degree they match. A policy engine is consulted before every request for an action or service.

The disassociation of security services from core service functionality increases the efficiency of Web services and reusability of security components.

Requirement 2: Layered security support

To enable interoperability of the security framework between multiple organisations, the proposed security framework will have three levels of compatible policies, namely, generic security policies, domain-specific security policies, and scenario-specific security policies. *Generic security policies* enable secure interoperability of multi-agent multi-domain (MAMD) systems between multiple organisations. The *domain-specific security policies* are applicable to a specific controlled environment or application domain. The *scenario-specific security policies* enable the specification of conventional and unconventional security constraints based on specific interaction requests. Supporting layered security allows secure collaboration and interoperability of Web services between organisations.

Requirement 3: Flexible, dynamic and adaptive

The heterogeneous, distributed and open nature of the Semantic Web dictates the security framework for the Semantic Web to be flexible, dynamic, and adaptive to different contexts. The distributed policy management allows entities to define their own security requirements and hence the autonomy to add, remove or adapt a policy to a specific scenario. The dynamic adaptation of policies is based on an entity’s credentials, interaction scenario, access criteria, available resources, and environment conditions. The use of a semantic-based approach to model security representations enables adaptive management of open systems (Tan & Poslad, 2004a). In this approach, the policy engine is capable of adapting a policy based on facts derived from the knowledge base which might involve environmental conditions. The advantage of having a flexible, dynamic and adaptive security framework is that Semantic Web entities would interoperate between multiple domains securely. The security framework itself would not be rigidly bound to specific security mechanisms and technologies.

Requirement 4: Semantically rich

The security framework for the Semantic Web needs to be able to describe and reason about security at the semantic level in order to provide access control at the finest granularity. The proposed security framework suggests a knowledge base component that will comprise a security ontology model and a process ontology model. The security ontology model is used to describe security concepts and their relations. The process ontology model is used to specify allowed interaction scenarios or business workflows. The use of semantic-based security representations supports the expressivity and analysability of security information. RDF and/or OWL will be used because of their high expressive power. The advantage of semantically rich representations is the ability to describe contexts at a higher level of abstraction, which is essential in reasoning and conflict resolution.

Requirement 5: Simple enough to automate

The security framework for the Semantic Web needs to be simple enough to automate so as to allow machines to access and process information securely. In order to simplify the automation of security services, the proposed security framework does not restrict entities on how to specify their policies. For instance, a policy may be as simple as a list of users and the services that they can or cannot access, or a policy may be a set of access right rules for specific users, service, and environment variables. The flexibility in policy specifications and the use of Semantic Web technologies such as RDF and OWL simplifies the process of automation of the security framework. The use of an automatic security framework enables autonomous systems to interoperate securely in an open environment without direct human intervention.

Requirement 6: Impervious to common network problems

The security framework for the Semantic Web needs to be resistant to problems inherent in the network and the distributed nature of its infrastructure. For the proposed security framework to be impervious to common security problems, it needs to incorporate existing security mechanisms and standards. The security mechanisms and standards component specifies instances of security concepts from existing security standards. The mechanisms and standards to be specified include communication security infrastructures such as SSL, PKI, X.509, etc. to ensure end-to-end network security. The advantage of this approach is that the framework will be able to utilise security mechanisms that have been proven to be useful and reliable over time.

Requirement 7: Implementable on current Semantic Web technologies

From the fact that this study does not involve the development or enhancement of the Semantic Web technologies, the proposed security framework needs to be implementable on the current adopted Semantic Web technologies. As discussed in Section 2.2, the technologies adopted or recommended by the W3C include Unicode, URI, XML, RDF, and OWL. For the proposed security framework to be implementable on current Semantic Web technologies, and in order to provide the intended flexibility, it is suggested that security concepts be defined in RDF and/or OWL. The Semantic Web is the future of the World Wide Web, therefore using Semantic Web technologies to implement the security framework will help in realising the envisioned benefits of the Semantic Web.

Requirement 8: Provide protection to all Semantic Web entities

The proposed security framework suggests the distributed policy management approach in order to provide security to all Semantic Web entities. Three components are suggested in this regard, namely *agent security*, *Web services security*, and *Web resources security*. In Web resources security, an external policy engine run by the Web server is used to perform security functionalities on behalf of the Web resources. In

Web services security, Web services specify their own security policies and include an internal policy engine that enforces security functionalities. In agent security, both agents and the agent platform need to be protected (Jansen, 2000). Agents specify policies that a requester must satisfy in order to use its service. Agents use an internal policy engine to enforce their policies. The advantage of using distributed policy management is the flexibility of the framework in using different approaches to secure different entities. Distributed policy management gives entities some degree of autonomy in determining which entities to trust in which situations.

Requirement 9: Provide a complete set of security services

The security framework for the Semantic Web needs to be able to provide a complete set of security services as discussed in Section 2.3. Providing a complete set of security services will avoid attackers using a weakness in one control to attack a different vulnerability. For instance, a weakness in authorisation process might result in a compromise in integrity. It is suggested that the proposed security framework should provide authentication, authorisation, integrity, confidentiality, non-repudiation and availability services. The security services component specifies security functionalities provided by the framework. The security ontology model relates the security services, threats and control mechanisms.

5.2.3. Concluding remarks

This section dealt with the adaptation of different components from the four selected existing security frameworks to fit the requirements of the security framework for the Semantic Web. The adaptation process involved selection of components that satisfy a particular requirement, or suggestions of modifications of certain components from existing security frameworks that are compatible with the identified requirement. The adaptation process assists this study in identifying components of the proposed security framework for the Semantic Web which will be discussed in the following section.

5.3. A SECURITY FRAMEWORK FOR THE SEMANTIC WEB

5.3.1. Introduction

This section presents the proposed security framework for the Semantic Web based on the adapted features discussed in the above section. The presentation of the proposed security framework starts with a summary of what the proposed security framework entails in Section 5.3.2. Section 5.3.3 presents the main components of the proposed security framework, and functionalities of the proposed security framework are presented in Section 5.3.4. An interaction scenario of the Semantic Web entities is presented in Section 5.3.5 to provide a behavioural view of the components of the proposed Security framework. Section 5.3.6 presents an implementation model and Section 5.3.7 presents the concluding remarks.

5.3.2. The proposed security framework for the Semantic Web

This section provides a (textual) primary presentation of the proposed security framework and an alternative (graphical) primary presentation. The textual presentation provides an overview of the proposed security framework. The graphical presentation uses a component-and-connector structure that shows how the framework is to be structured as a set of elements that have runtime behaviours and interactions (Bass et al., 2003).

The proposed security framework for the Semantic Web integrates three domains, namely: 1) Semantic Web technologies, 2) information security, and 3) Web services. Figure 5.1 below illustrates the integration of different aspects of the three domains.

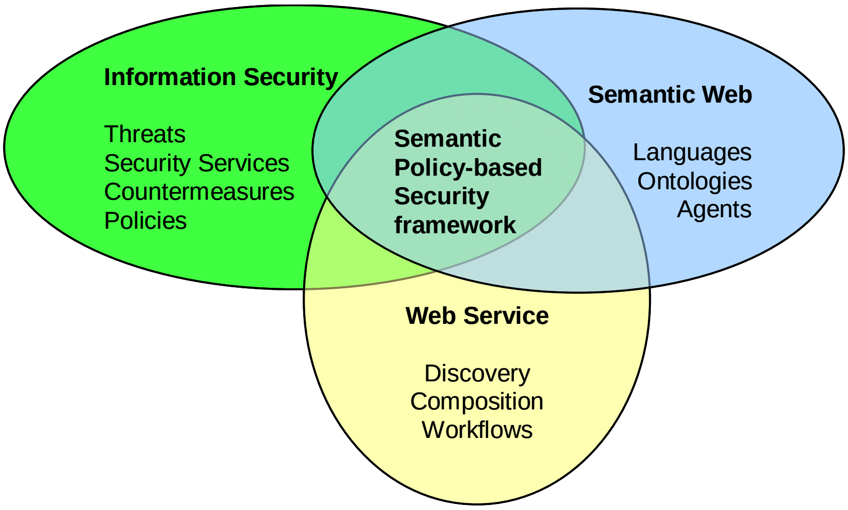


Figure 5.1: Semantic policy-based security framework

This study proposes a policy-based security framework for the Semantic Web that is flexible, dynamic and adaptive to different contexts. The proposed framework is semantically rich and uses a distributed policy management approach to specify and enforce security policies.

The framework makes use of ontologies to describe security information in order to support secure transactions across heterogeneous multi-domain boundaries. The use of the Semantic Web technologies in implementing the proposed framework simplifies the process of automating the security services provided by the framework. Describing the semantics of security concepts enables reasoning about security-related information in an intelligent manner. In the proposed security framework, a reasoning engine reasons about the facts from the knowledge base of security concepts and the security policies to make an informed decision on security services to provide and security mechanisms to use.

The advantage of policies based on semantics of security ontology, where the ontology provides a limited set of nouns to associate facts as rules and policies, lies in providing additional support to cluster policies and so optimise management. The advantage of the semantically rich representation of policies lies in allowing description of contexts and associated policies at a high level of abstraction that enables their classification and comparison in order to detect conflicts between policies (Bhargavan, Fournet & Gordon, 2004).

From the adaptation process discussed in Section 5.2 above, the proposed security framework for the Semantic Web will have four main components as illustrated in Figure 5.2 and discussed above.

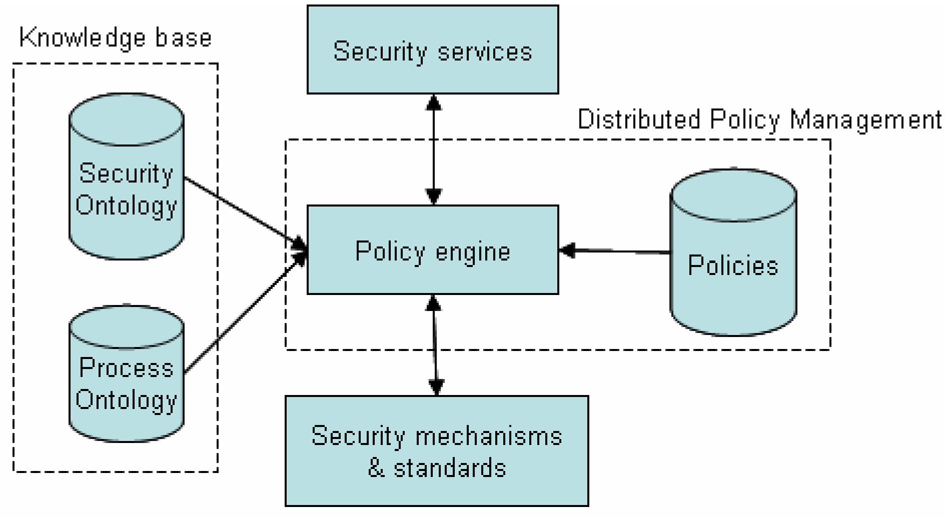


Figure 5.2: Components of the security framework for the Semantic Web

In the following sections, different views of the proposed security framework are presented in order to provide a coherent understanding of features, functionalities and the interaction of the framework's components.

5.3.3. Components of the proposed security framework

This section presents the *elements and their properties* section of the element catalogue. A detailed explanation of the components (elements) of the proposed security framework is presented below.

5.3.3.1. The distributed policy management

In distributed policy management each Semantic Web entity specifies its own security policies and uses a policy engine to enforce its policies

(Kagal et al., 2003). The framework adopts semantically rich representations for policy definitions.

Policy engine

The policy engine provides reasoning features needed to deduce new information from existing knowledge. A policy engine is used as a reasoning engine and is consulted before every request for an action or service and before taking any action. A policy engine may be internal to a Web entity, in the case of agents and Web services, or external in the case of Web resources. A policy engine draws facts and rules from the knowledge base about security concepts and compares these facts with the applicable security policy of the requested entity to determine a course of action.

Policies

The proposed security framework suggests a set of flexible and adaptive security policies to enforce modular generic security services in an open multi-agent multi-domain environment. The need for flexible policies that ensure security and timely information processing depending on the application and the particular scenario is emphasised by Thuraisingham (2005). A policy model defines a set of policies which can be applied to entities that need to interact with protected entities. From the discussion of existing security frameworks in Section 2.4, several security policies could be used by the proposed security framework to address security threats to the Semantic Web. The adopted security policies include identification policy, access control policy, privacy policy, transactions policy, threat-countermeasure policy, and spatial policy.

Identification policy

The identification policy specifies the type of authentication mechanism(s) that may be used in different contexts. For instance, a policy may specify that a PKI-based credential be used for entity identification. The identification policy is used to enforce authentication service to Semantic Web entities.

Access control policy

The access control policy specifies entities to be protected, against whom (requesting entities), and what mechanisms are to be used for protection (Ventuneac et al., 2003). The framework supports the specification and computation of access control policies for composite Web services as outlined in the desiderata for access control mechanisms for the Semantic Web services by Agarwal and Sprick (2004). The access control policy is used to enforce authorisation service upon Semantic Web entities. The access control policy also supports confidentiality by ensuring authorised access to Web entities. The separation of identification policy from the access control policy allows virtual association between interacting entities, and provides a basis for privacy (Demchenko et al., 2005).

Privacy policy

The privacy policy specifies an entity's privacy preference regarding the use of its data, with whom they are to be shared, and for how long the data may be retained (Turner et al., 2005). The privacy policy for Web service may be specified by extending the OWL-S service profile input parameters to include *purpose*, *recipient*, and *retention* properties. The purpose property describes the reasons for collecting users' data. The recipient property declares the entities with whom the data will be shared. The retention property defines the activity scope during which the data will be retained. The privacy policy is also used to enforce the confidentiality service to Web entities.

Transactions policy

The transactions policy specifies security mechanisms to provide a trusted environment within which to conduct business processes. The transactions policy ensures integrity, availability and non-repudiation of transactions between Web entities. The transactions policy is also used to ensure the correct order of execution of processes and services in composite Web services by extracting relevant information from the process ontology model.

Threat-countermeasure policy

The threat-countermeasure policy specifies which type of countermeasure (security mechanism) is needed to provide protection against specific security threats. The threat-countermeasure policy is used to specify protection in the case of common network problems such as communication security of Web resources. For instance, a threat-countermeasure policy may specify that SSL (security mechanism) be used in all communications to a Web service (protected entity) to provide integrity (security service) to avoid session hijacking (threat).

Spatial policy

The spatial policy specifies constraints based on environment conditions derived from the knowledge base. The spatial policy is used to specify non-conventional security requirements of Web entities. It provides contextual information that can be used in access control of resources. The use of contextual information provides an active security model (Toninelli et al., 2006) that is aware of the context associated with the ongoing activity in providing access control and simplifying access control management. An active security model increases policy specification reuse and makes policy update and revocation easier. Factors such as time, location, and so on specified in spatial policy may be used to constrain an entity's behaviour.

A semantic language is necessary to describe information in a well-defined and structured manner so that machines, rather than humans, can read and understand it (Joshi et al., 2002). The proposed security framework may describe policies in RDF-S or OWL and allows the inclusion of component policies that use different policy expression formats such as XACML, Rein, etc. The security for the Semantic Web entities is illustrated in Figure 5.3 below.

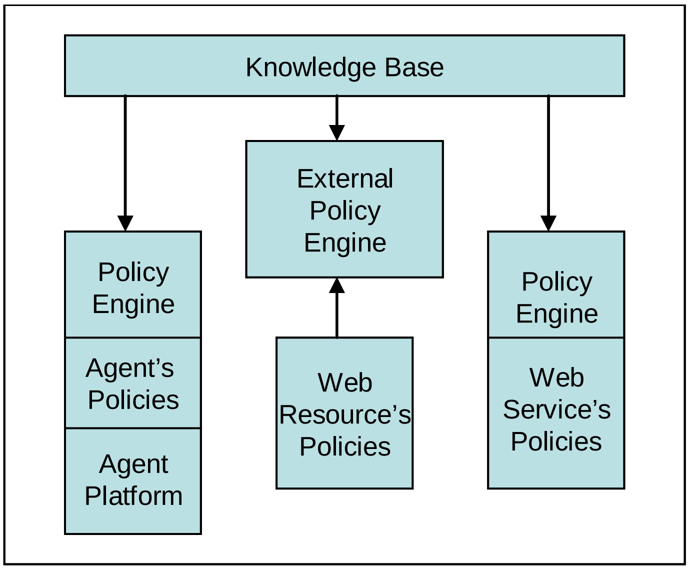


Figure 5.3: Distributed policy management

Agent security

Agent security involves two concepts, namely, protecting the agent from malicious hosts and other agents, and protecting the agent platform from malicious agents (Jansen, 2000). For the purpose of this dissertation, agent security entails the former description. Security aspects related to agents are discussed in Section 2.3.2.1. In the proposed security framework, agents are annotated with security concepts defined in the security ontology. Security concepts annotated to agents include functional capabilities, functional requirements, security capabilities, and security requirements. An agent defines its own security policies and uses an internal policy engine to enforce its security policies.

Web services security

Web services security ensures secure execution of Web services with proper authorisation to access a particular Web service. Security aspects of the Web services were discussed in Section 2.3.2.2. Web services are annotated with security concepts defined in the process ontology. Security concepts annotated to Web services include security requirements, capabilities, preconditions and effects. Web services define their own security policies and use an internal policy engine to enforce their security policies. Each request to a Web service passes through the Web service's policy engine, which then fetches relevant security information from the knowledge base and the Web service policies before it can reason about the request.

Web resource security

Web resources security involves the protection of resources such as Web pages, XML documents, etc. as discussed in Section 2.3.2.3. Web resources are annotated with concepts defined in the security ontology. Concepts annotated to Web resources include security requirements and capabilities. The Web resource defines its own security policies but uses an external policy engine run on a Web server to enforce its security policies. The Web server is trusted to perform security functionalities on behalf of the resource. The external policy engine gets its input from the Web resources' policies, the knowledge base, and the requesting entity and reasons about the request on behalf of the Web resources. The use of an external policy engine for Web resources is suggested for two reasons. Firstly, most of the Web resources are not capable of enforcing their own security policies. Secondly, this approach prevents the requesting entities from having access to a Web resource before the authorisation decision is made. If an internal policy engine were to be used, the requesting entity would have some sort of access to the Web resource before the Web resource could reason and make a decision to grant or deny access.

5.3.3.2. The knowledge base

This component involves the description of relationships between protected entities and protected mechanisms that are used to provide specific security services. The knowledge base comprises the security ontology model and the process ontology model.

Security ontology

Security ontology models are used to describe security concepts and their relations. Security ontologies provide abstract means for capturing security concepts and explicit means for liaising between security specifications (Tan & Poslad, 2004a). Ontologies are the basis for performing automatic reasoning about security annotations. The framework provides ontologies for describing policy and mechanisms for reasoning about them. Languages such as RFD-S and OWL help the Web to preserve maximum expressiveness of local policies by enabling global interoperability of policy reasoning (Kagal et al., 2006). By using semantic models such as RDF-S, entities can interpret security information more correctly. An ontological model promotes the interoperability between disparate security systems (Tan & Poslad, 2004b). Figure 5.4 below illustrates the security ontology model used in the proposed framework.

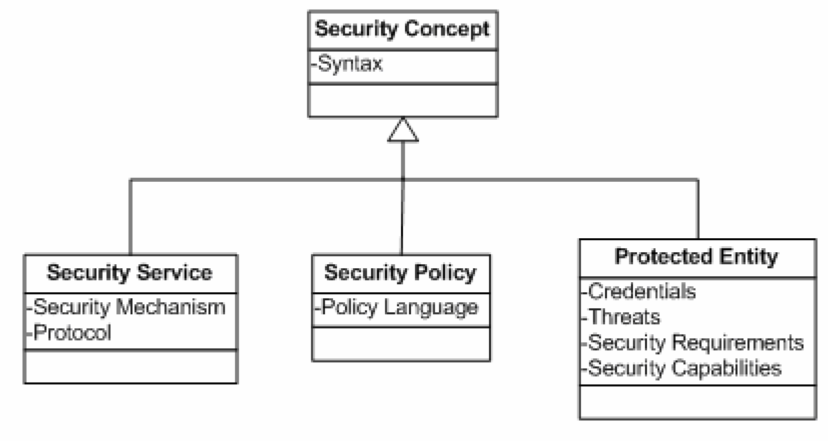


Figure 5.4: Security ontology model class hierarchy

Process ontology

Process ontology models are used to specify allowed interaction scenarios or business workflows. A *business process* can be defined as a sequence of activities with distinct inputs and outputs that serve a meaningful purpose within or between organisations (Singh & Salam, 2006). The process ontology model provides a means to constrain the invocation of business processes and Web services. It provides a detailed service process description of the workflow. To provide semantic service discovery (Jiang, Chung & Cybenko, 2003), the framework uses semantic description and semantic matching that allow the service discovery requester to specify additional constraints on attributes such as priority, matching rule, and so on. The proposed security framework will utilise the OWL-S process model (Ankolekar et al., 2001; Denker, Nguyen & Ton, 2004) to describe a common ontology for services. OWL-S is a specification for the semantic description of Web services that facilitates their automation (Howard & Kerschberg, 2004). The ontology enables *reactive* service composition (Joshi et al., 2002) by dynamically discovering, integrating, and executing individual services available in the environment. Figure 5.5 below illustrates the process ontology model.

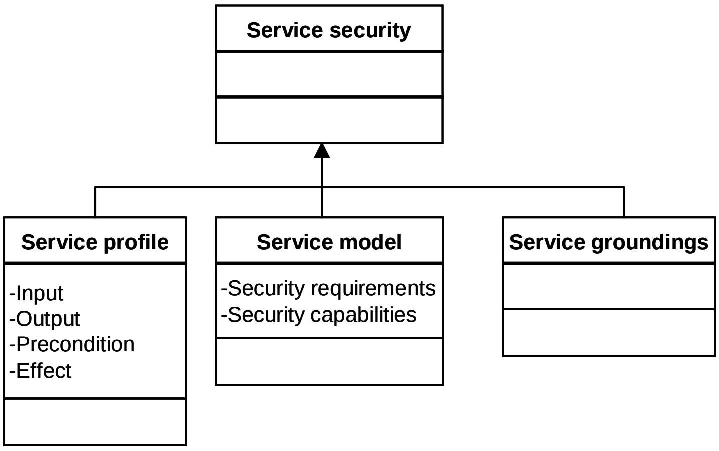


Figure 5.5: Process ontology model class hierarchy

5.3.3.3. Security services

The security services component provides an abstract means of defining security functionalities to be provided by the framework. Security services required for a security framework for the Semantic Web include authentication, authorisation, integrity, confidentiality, availability and non-repudiation. There is a discussion of security services for the Semantic Web in Section 2.3.4, Section 4.4.2 and Section 4.5.2.9. Capturing security services in a high-level conceptual model in a semantic-based approach allows the mapping of security services to existing security specifications. Security services provide various protections against security threats to specific entities. For instance, integrity service provides protection against interruption and modification threats to Web resources. Binding of core services and security services can be defined dynamically at the moment of service invocation by using existing Web services or

XML security technologies for binding security services and policies (Demchenko et al., 2005).

5.3.3.4. Security mechanisms and standards

This component specifies instances of security concepts, policies, and service entities based on existing security standards as discussed in Section 2.4. Security mechanisms and standards are also considered as essential components of a security framework as presented in Section 4.4.4. This component allows the incorporation of existing XML-based security standards that can be adapted to semantic-based applications. Security standards such as XML encryption, XML Signature, XACML, XKMS, and SAML may be utilised by the security framework to provide different security functionalities.

5.3.4. Functionalities of the proposed security framework

This section presents the *relations and their properties* section of the element catalogue. The components (elements) of the proposed security framework provide clearly defined functionalities (relations). The functionalities provided by the proposed security framework are discussed below.

5.3.4.1. Description of security concepts

The proposed security framework provides a means of describing security concepts in a structured way that allows reasoning through this information. The description of security concepts is done through the use of the security ontology model. The description of security concepts allows entities to describe their security requirements and capabilities.

5.3.4.2. Description of service workflow

The process ontology model of the proposed security framework allows the description of the service workflow. The description of service workflow involves the description of allowed interaction scenarios, service composition, and execution order.

5.3.4.3. Policy specification

The proposed security framework uses the distributed security management approach in which each entity is capable of specifying its own security policies. Policy specification involves the specification of security rules and constraints. Policies are based on the semantics of a security ontology which provides a limited set of nouns to associate its facts as rules in policies. Policy expressions can also be based on axioms defined using terms from the abstract security ontology. The framework supports the use of standard declarative semantic languages for expressing policies. Security policies include general rules for protecting entities, detecting threats, and countermeasures. The security policies supported by the framework are classified as *identification policies*, *access control policies*, *privacy policies*, *threat-countermeasure policies*, *transactions policies* and *spatial policies*. The spatial policies deal with environmental conditions and temporal information from the knowledge base. The threat-countermeasure policies specify which type of countermeasure is needed to protect against certain threats. Policy specification is a major step in defining a system that is able to protect Web resources. Semantic Web languages such as RDF or OWL are used to specify security policies.

5.3.4.4. Reasoning about security concepts and policies

The proposed security framework provides a means by which reasoning can take place about applicable access rights, prohibitions and other security concepts. A policy engine performs reasoning on behalf of Web entities. The policy engine is therefore consulted before every request for action or service and before taking any action. The reasoning functionality is performed based on the facts from the knowledge base loaded from the security ontology and on rules extracted from the policy ontology.

5.3.4.5. Specification of security services

The proposed security framework supports the specification of specific security services such as integrity, confidentiality, and so on for a particular Web service or Web resource. Security services are specified by security policy that provides a mapping between the request context and applicable permissions. Security services may be bound to, and requested from, Web services using standard request/response format that uses existing Web services and XML technologies for binding security services and policies to Web service description.

5.3.4.6. Specification of security mechanisms and standards

The proposed security framework supports the specification of specific instances of security concepts, services, and policies. The specification of specific instances of security concepts allows the incorporation of XML-based security standards to provide specific security services.

Table 5.1 below summarises the main functionalities of the proposed security framework for the Semantic Web.

Table 5.1: Functionalities of the proposed security framework

Component	**Functionality**
Security ontology model	Description of security concepts
Process ontology model	Description of service workflows
Policy model	Policy specification
Policy engine	Reasoning about security concepts and policies
Security services	Specification of security services
Security mechanisms and standards	Specification of security mechanisms and standards

5.3.5. Interaction views of the proposed security framework

This section presents the *element behaviour* section of the element catalogue. A behaviour description provides information that reveals the ordering of the interactions among the components, concurrency, and time dependencies of the interactions (Bass et al., 2003). Interaction diagrams model dynamic behaviour and are used to describe the interaction between objects and messages in a system within a single use case (Avison & Fitzgerald, 2006). In the context of the proposed security framework, element behaviour describes the interaction of the Semantic Web entities within the proposed security framework. A common scenario is described, namely, a software agent requesting access to a Web service.

Agent requesting access to a Web service

This scenario starts with an agent who has discovered a Web service and requests access to the discovered service. The agent is not known by the Web service a priori. The agent submits its credentials including security requirements and capabilities. Upon receiving the request, the Web service uses its internal policy engine to check whether the agent meets the security requirements of the Web service. The policy engine draws information from the security ontology, service ontology, and policy ontology to decide whether to grant access to the agent or not. If the access is to be granted, the policy engine invokes appropriate security mechanisms to enforce the specified security service. Figure 5.6 below illustrates this scenario.

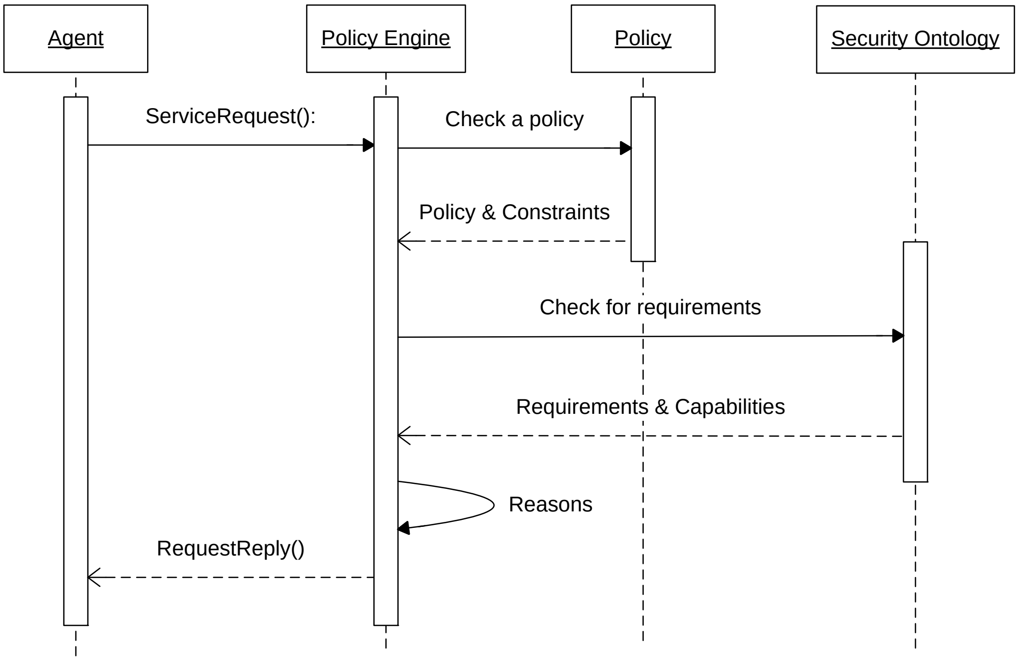


Figure 5.6: Interaction of the framework components and Web entities

5.3.6. Implementation model of the proposed security framework

Similar to the logic and physical levels of system abstraction by Avison and Fitzgerald (2006), are essential and implementation models by Whitten, Bentley and Barlow (1994).

For the purpose of the proposed security framework, an implementation model or a physical level abstraction is necessary owing to the requirement that the security framework be implementable by using Semantic Web technology. The implementation model will depict the technologies to be used in implementing the framework. The model will make it possible to analyse the relationships between technologies used and their interoperability.

Language technologies to be used in the implementation of the proposed security framework include *URI, Unicode, XML, RDF(S)*, and *OWL*. To obtain a secure Semantic Web these technologies need to be secured (Thuraisingham, 2002). The Semantic Web entities use different technologies for their implementation, hence the need for different technologies for their security including *agent security*, *Web service security*, and *Web resource security*. The discussion of agent security, Web service security and Web resource security was presented in Section 5.3.2 of this dissertation.

- *Unicode/URI* allows data and texts to be exchanged globally between different systems. The unique identification mechanisms provided by Unicode/URI ensure availability of Web resources in different platforms. This layer also supports identity verification, which can be used to provide non-repudiation of Web transactions.
- *XML security* involves the use of XML technology in providing security services as well as securing XML documents. XML security includes XMLEnc for end-to-end encryption of XML-based documents, XMLDSig for signing and verification of entities, and XKMS for key distribution and management.

- *RDF security* involves securing RDF documents and the use of RDF-S in describing basic security semantics of Web entities. RDF can be used to specify and enforce security policies (Carminati et al., 2004).
- *OWL security* involves securing ontologies and descriptions of security concepts and policies by using OWL. Subcomponents of OWL security are the security ontology, the OWL-S process ontology, semantic-based security policies, and the policy engine. *Semantic-based policies* involve the use of OWL in specifying security policies. The *policy engine* uses OWL to reason about security concepts and policies and for inference control (Xu, Li, Lu & Kang, 2006).
- *Conventional security solutions* include security standards and mechanisms such as SSL, X.509, etc. that can be invoked by different security technologies depicted in the implementation model. Incorporation of conventional security solutions provides protection to common network problems.

The issue of secure XML documents, secure RDF documents, and secure OWL documents implies the need for access control mechanisms for each layer of the Semantic Web architecture. Access control for XML, RDF and ontologies protects layers of the Semantic Web as directed by Thuraisingham (2005).

Figure 5.7 below provides a pictorial view of the implementation model for the proposed security framework for the Semantic Web. The model indicates the Semantic Web technologies to be used for the implementation of the proposed security framework.

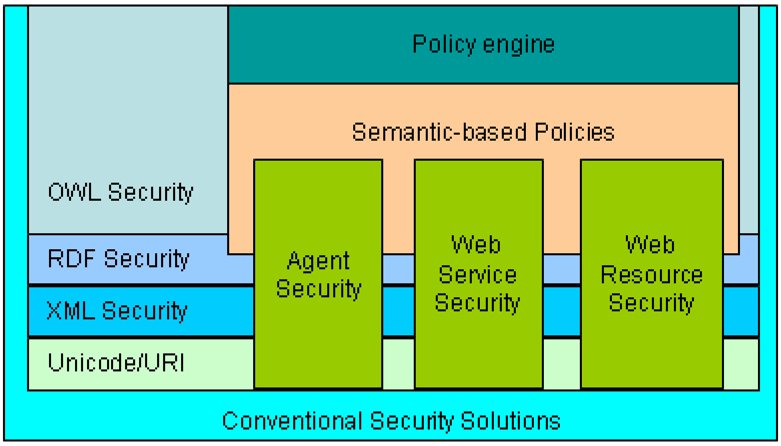


Figure 5.7: Implementation model of the proposed security framework

From the implementation model described above, one can outline a security stack of the Semantic Web (Gerber, 2007) by arranging security functionalities of the Semantic Web technologies depicted on the model in a layered approach.

Table 5.2 below summarises security functionalities according to the Semantic Web layers.

Table 5.2: Security stack of the Semantic Web

Security technology	**Security functionality**
Unicode/URI	Identity verification
XML security	Secure XML documents, encryption, key management, and digital signature
RDF security	Secure RDF documents and description of basic security semantics of Web entities
OWL security	Secure OWL documents, description of security ontology, security policies, inference control, and reasoning of security concepts

The security stack does not show security services such as integrity, authentication, or confidentiality; rather it shows security functionalities provided in different language technologies. The security functionalities can be used to provide several security services as specified for a particular application context in the security ontology and policies.

5.3.7. Concluding remarks

The proposed security framework for the Semantic Web is compiled by integrating components adapted from four existing security frameworks discussed in Section 4.6. By integrating all the components adapted to the requirements of a security framework for the Semantic Web, the proposed security framework will adhere to the requirements of a security framework for the Semantic Web. The proposed security framework contains all the essential components of a security framework identified in Section 4.6. The documentation approach of the proposed security framework is summarised in Table 5.3 below.

Table 5.3: Documentation approach for the proposed security framework

Primary presentation	Textual	Section 5.3.2
	Graphical	Section 5.3.2; Figure 5.2
Element catalogue	Elements and their properties	Section 5.3.3
	Relations and their properties	Section 5.3.4
	Element behaviour	Section 5.3.5; Figure 5.6
Physical abstraction	Implementation model	Section 5.3.6; Figure 5.7

5.4. EVALUATION OF THE PROPOSED SECURITY FRAMEWORK

In this section, the proposed security framework for the Semantic Web is evaluated against the requirements of a security framework discussed in Section 4.4. The evaluation involves the description of how each requirement is satisfied by the proposed security framework. Components used to satisfy a particular requirement are also mentioned.

Decoupling of security functionality from core service functionality

The framework uses a policy engine or reasoning engine to enforce security services. Specification of security services on the framework allows the framework to deal with security functionalities while a particular Semantic Web entity deals with its core functions.

Layered security support

The use of compatible *generic security policies*, *domain-specific security policies*, and *scenario-specific security policies* allows the framework to support layered security.

Flexible, dynamic and adaptive

The use of distributed policy management provides entities with the autonomy to add, remove, or adapt a policy to a specific scenario. The dynamic adaptation of the framework is facilitated by the use of spatial policies whose constraints may change based on environmental conditions.

Semantically rich

The proposed security framework is semantically rich as it provides security at Semantic level by using technologies such as RDF, RDF-S, OWL, and OWL-S. The use of security ontology and process ontology allows reasoning about security concepts.

Implementable on the current Semantic Web technologies

The framework is implementable on the current Semantic Web technologies as indicated in the implementation model of the proposed security framework. The framework uses Semantic Web technologies such as Unicode, URI, URL, XML, RDF, RDF-S, OWL, and OWL-S.

Simple enough to automate

As discussed in Section 5.2.3, the flexibility in policy specifications and the use of Semantic Web technologies simplifies the process of automation of the security framework. The framework is simple enough to automate because it uses languages that are understandable by machines.

Impervious to common network problems

The proposed security framework is considered to be impervious to common network problems as it incorporates conventional security solutions. Conventional security solutions involve security standards and mechanisms that provide communication security, end-to-end security, key management and distribution, and so on.

Provides protection to all Semantic Web entities

The framework provides security to all Semantic Web entities, i.e. agents, Web services, and Web resources as discussed in Section 5.2.3. The use of distributed policy management supports protection to each Semantic Web entity.

Provides a complete set of security services

The framework provides a complete set of security services, i.e. authentication, authorisation, integrity, confidentiality, availability and non-repudiation, as discussed in Section 5.2.3.

From the above evaluation of the proposed security framework and the adaptation of existing security frameworks to the requirements of a security framework for the Semantic Web, it is argued that the proposed security framework satisfies all the requirements of a security framework for the Semantic Web. Table 5.4 below summarises the evaluation of the proposed security framework in relation to the requirements of a security framework for the Semantic Web.

Table 5.4: Evaluation of the proposed security framework

Requirement	**A proposed security framework for the Semantic Web**
Decoupling of security functionalities from core service functionalities	**Satisfied** by the use of policy engine and specification of security services
Layered security support	**Satisfied** by the use of generic, domain-specific, and scenario-specific security policies
Flexible, dynamic and adaptive	**Satisfied** by the use of distributed policy management
Semantically rich	**Satisfied** by the use of security ontology and process ontology
Simple enough to automate	**Satisfied** by the use of Semantic Web technologies
Impervious to common network problems	Satisfied by the incorporation of security standards and mechanisms
Implementable on current Semantic Web technologies	**Satisfied** by the use of XML, RDF, OWL, RDF-S, and OWL-S
Provides protection to all Semantic Web entities	**Satisfied** by the use of distributed policy management where agent, Web service, and Web resource are protected entities
Provides a complete set of security services	**Satisfied** by the use of security services component where security services are specified

5.5. CONCLUSION

The development of a security framework for the Semantic Web was presented in this chapter. The approach used to develop the proposed security framework is to adapt certain components from the existing security framework to the requirements of a security framework for the Semantic Web, and then to integrate the adapted components. Section 5.2 presented the adaptation process and Section 5.3 presented the proposed security framework for the Semantic Web.

The study proposes a security framework for the Semantic Web that uses a distributed policy management approach to enable each of the Semantic Web entities to specify and enforce its own security policies. The proposed security framework makes use of Semantic Web technologies to enable interoperability of security components in multi-agent multi-domain environments. The proposed security framework is flexible, dynamic and adaptive to different contexts and application domains. The functionalities of the proposed security framework include the *description of security concepts*, *description of service workflow*, *policy specification*, *reasoning about security concepts and policies*, *specification of security services*, and *specification of security mechanisms and standard*s. The implementation model enables the analysis of the Semantic Web security technologies to outline the security stack of the Semantic Web by arranging the security functionalities of language technologies in a layered approach.

The proposed security framework satisfies all the requirements of a security framework for the Semantic Web as shown in the evaluation of the proposed security framework.

In the following chapters different usage scenarios are presented as a proof-of-concept to strengthen the research results presented in this chapter.

CHAPTER 6: PROOF-OF-CONCEPT SCENARIOS

6.1. INTRODUCTION

In this chapter, the proposed security framework for the Semantic Web is applied to different usage scenarios as a proof-of-concept. The application scenarios illustrate major aspects of the proposed security framework for the Semantic Web. The focus of the application scenarios is to explain further how the proposed security framework for the Semantic Web works and the functionalities it provides.

The framework application of a particular case scenario will start by giving a brief description of the scenario followed by the outline of security functionalities required by the scenario. The scenarios are explained chronologically as the sequence of events that take place during the scenario execution. The description of how the framework provides the required functionalities presents the actual application of the framework to the scenario as a proof-of-concept.

Section 6.2 presents the application of the proposed security framework for the Semantic Web to different usage scenarios, while Section 6.3 concludes the chapter by summarising the findings from the chapter.

6.2. APPLICATION SCENARIOS

The scenarios used in the application of the proposed security framework have been extracted from the existing body of literature. The scenarios selected highlight several security aspects that have to be dealt with in the proposed security framework. On applying the proposed security framework to the scenarios the focus will be on the security functionalities supported by the proposed security framework and the components used to provide the functionality.

6.2.1. Scenario 1: Basic interaction scenario

The basic interaction scenario presented by Ashri et al. (2004) highlights security issues arising when two service providers need to interact to achieve a client's goal. In this scenario a client A needs to make use of Grid Compute Service (service provider B) in order to perform a calculation, but requires that the data for the operation to be provided to B by a Data Service (service provider C). Each party in this scenario belongs to a different organisation and as a result may have different security requirements and capabilities. B and C have no prior knowledge of each other's existence.

The interaction starts from the point where client A has discovered B and C as the service providers it wishes to use in order to achieve its computing task. A requests C to allow B to retrieve the relevant data. C notifies A that the data are at B. A requests B to run the calculation on the data. Lastly, A is notified by B that the calculations have been finished and that the results are ready. The scenario is illustrated in Figure 6.1(a) below.

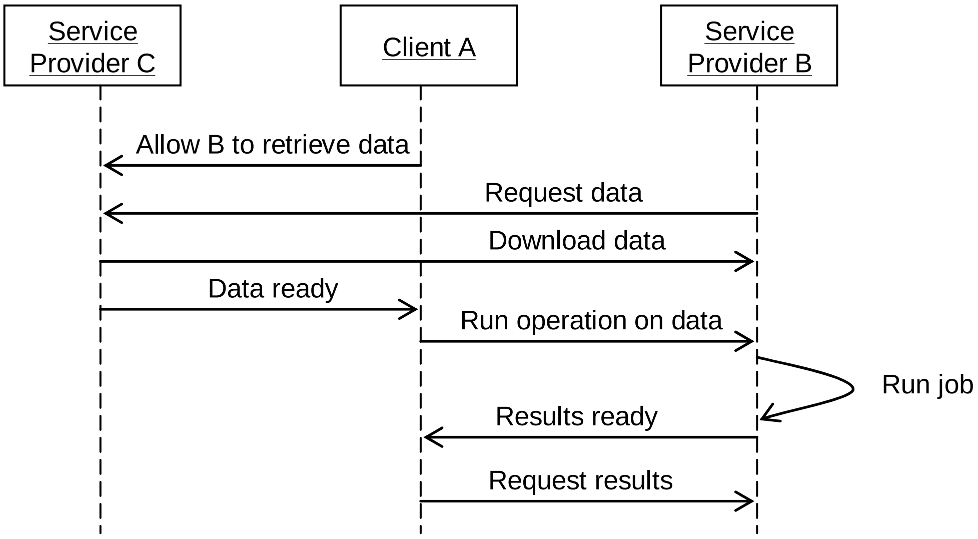


Figure 6.1(a): Basic interaction scenario

The interacting parties in the scenario described above are from different organisations, hence the need to be able to describe domain-based security requirements and capabilities. Since the parties do not know one another a priori, there is a need for the interacting parties to identify one another based on their security policies. The interacting parties may need a guarantee that their data are not misused or passed to other parties, hence the need for integrity and confidentiality services. The service providers may need to know whether the requested interaction is allowed, which leads to the necessity for being able to describe allowed interactions or workflows and reasoning about the requested interactions.

On applying the proposed security framework to the above scenario, the interacting parties may describe their security requirements and capabilities by using the *security ontology*. In the security ontology, the protected entity, a subclass of the security concept, has security requirements and security capabilities as attributes. These attributes of the protected entities are used to specify the security requirements and security capabilities of the interacting parties.

The *identification policy* may be used by both service providers to specify the mechanism to be used to identify the requesting party in different contexts. Based on the requester’s credentials, specific authentication mechanisms may be invoked. For instance, service provider C may need client A to provide a digital certificate before data can be released to service provider B. The identification policy may also be used to specify whether the requesting party should be authenticated or not before it can be allowed to interact with the service provider. For instance, if A and C are from the same domain, C may release data to B without authenticating A.

Upon receiving the request to supply data to service provider B, service provider C may use its *policy engine* to reason as to whether service provider B can enforce integrity and confidentiality on the requested data. The security capabilities of service provider B will determine the security service it may enforce. For instance, if the service provider B is capable of encrypting data, then it can be considered as capable of enforcing integrity service.

The *process ontology* can be used to specify the allowed interactions. The OWL-S process model ontology allows the use of preconditions and effects to describe aspects of interactions. The policy engine of the service provider C may use the information from the process ontology to decide whether to allow the interaction or not.

Components of the proposed security framework used in this scenario include the security ontology, identification policy, policy engine, and the process ontology. Functionalities of the proposed security framework illustrated by the scenario include *description of security concepts* (security requirements and capabilities*), policy specification* (identification policy), *reasoning about security concepts* (policy engine), and *specification of allowed interactions* (process ontology). Figure 6.1(b) illustrates the application of the proposed security framework to the scenario.

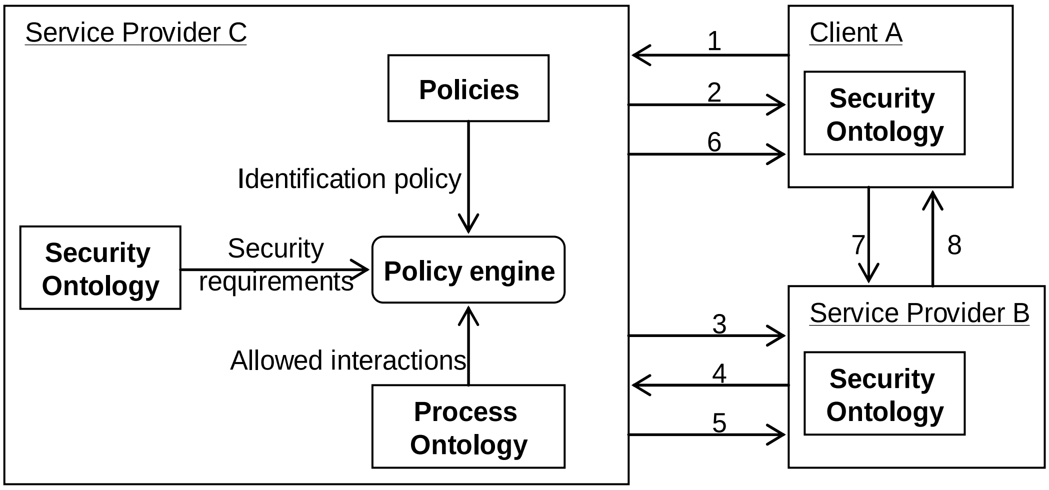


Figure 6.1(b): Secure basic interaction scenario

The secure basic interaction scenario starts with client A requesting service provider C to allow service provider B to download data required for some calculations (step 1). The policy engine of the service provider C gets security requirements and the identification policy to determine whether to authenticate client A and service provider B before releasing data. The service provider C then authenticates client A (step 2) before authenticating service provider B (step 3). Once service provider B has been authenticated, the security capabilities of the service provider B are sent to service provider C (step 4). Upon receiving the security capabilities of service provider B, service provider C reasons about the requested interaction to determine whether it is allowed. Allowed interactions are fetched from the process ontology. If the interaction is allowed and service provider B meets the security requirements of service provider C, then data will be released (step 5). In step 6 service provider C notifies client A that data are available at service provider B. Client A will then request service provider B to perform calculations on the data (step 7). Lastly, service provider B notifies client A that the results are ready (step 8).

6.2.2. Scenario 2: Travel Web service scenario

A scenario presented by Denker et al. (2004) describes a situation where agents and Web services have security mark-up as well as other functionally oriented mark-up. An agent has to find a Web service that provides specific functionalities and fulfils certain security constraints.

The scenario starts with an agent A looking for a travel Web service. Agent A submits a request for a Web service describing the desired functionalities of the Web service as well as the agent's security requirements and capabilities. The assumption is that the agent is only capable of performing Open-PGP encryption and requires that the travel service be capable of authenticating itself and communicating in XML. The travel service is capable of using XKMS protocols for message exchanges and it requires an agent to be able to perform encryption. The agent wants a travel service that is able to reserve a flight, buy a ticket, rent a vehicle, and book a hotel room. The travel Web service is the composition of transportation service, accommodation service and entertainment service. The travel Web service therefore needs to use the air transportation service to reserve a flight and buy a ticket, the land transportation service to rent a vehicle, and the accommodation service to book a hotel room. The scenario is illustrated in Figure 6.2(a) below.

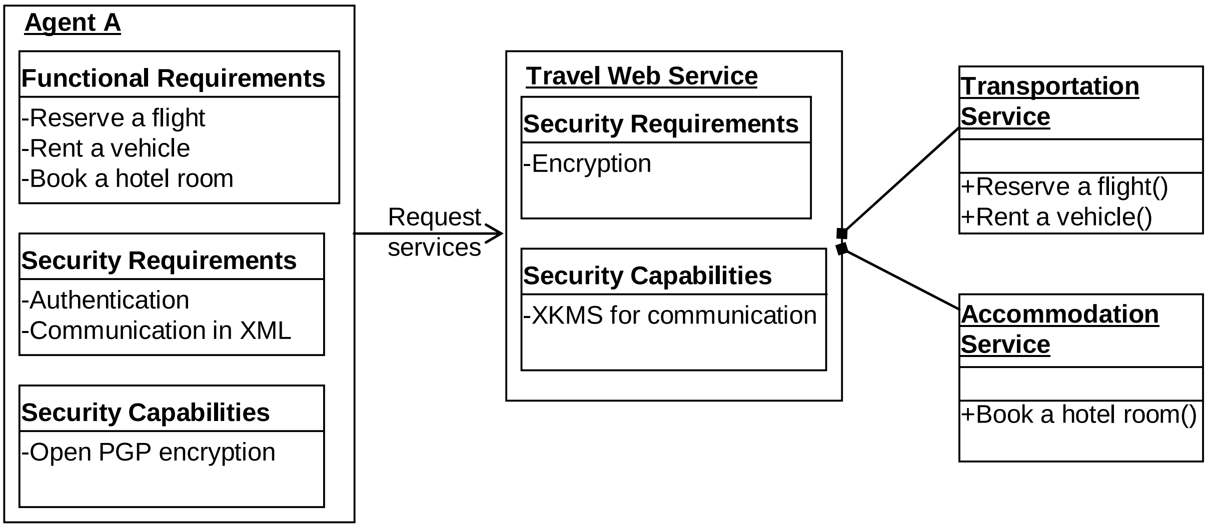


Figure 6.2(a): Travel Web service scenario

The main security concern in this scenario is the mark-up of security requirements and capabilities of the interacting parties. Matching of the security requirements of the requesting entity and the security capabilities of the requested entity is another important and challenging issue. Secure composition of services to achieve a desired task is another security aspect required by the scenario. For instance, on buying an air ticket there are issues of secure communication of financial and personal information from the client to the service provider, as well as the issue of non-repudiation. On booking a hotel room there may be non-conventional constraints such as ‘book a hotel room only if the hotel is near a shopping mall’.

On applying the proposed security framework to the above scenario, the interacting parties will describe their security requirements and security capabilities through the use of *security ontology*.

The *policy engine* can be used to implement a matching algorithm to decide whether the security requirements of the requesting party match the security capabilities of the requested party.

To ensure secure composition of service, the policy engine can be used to ensure that the preconditions of a particular service are met by getting relevant information from the *process ontology*. The process ontology is used to constrain the invocation of Web services.

The requesting party may specify a policy that requires the enforcement of integrity and confidentiality services for every financial transaction. The *transactions policy* may be used to specify the security services required for a transaction. The requested party may specify a transaction policy that enforces non-repudiation service for every financial transaction.

Non-conventional constraints may be specified by using *spatial policies*. There may be a spatial policy that specifies non-conventional constraints such as ’book a room only if the hotel is within a specific location’.

The components of the proposed security framework used in this scenario include security ontology, process ontology, transaction policy, spatial policy, and the policy engine. The functionalities provided by the proposed security framework to the scenario include *description of security concepts* (security ontology), *reasoning about security concepts* (policy engine), *secure composition of services* (process ontology), and *policy specification* (transaction policy and spatial policy). Figure 6.2(b) illustrates the application of the proposed security framework to the scenario.

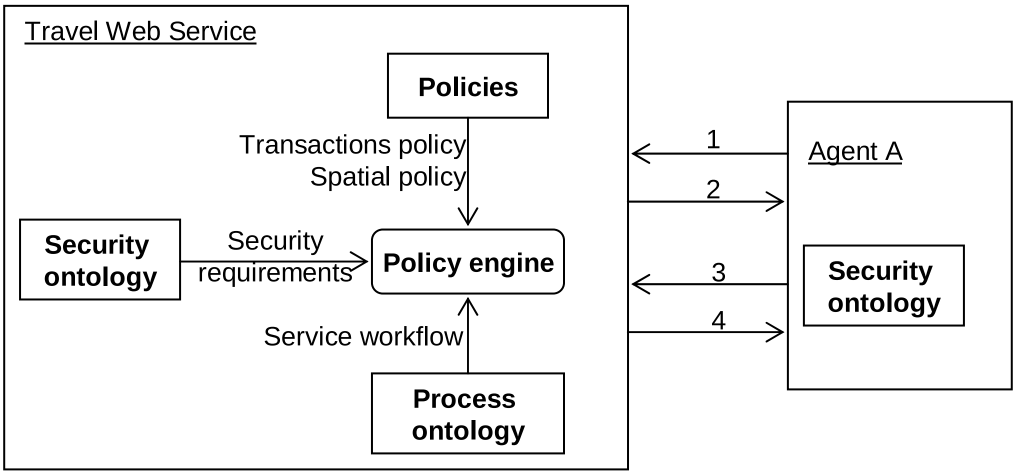


Figure 6.2(b): Secure travel Web service scenario

The secure travel Web scenario starts with agent A requesting services from the travel Web service (step 1). Upon receiving the request, the travel Web service fetches its security requirements and policies (transaction policy and spatial policy) and reasons about the request to determine whether it can fulfil the request. It then requests the security capabilities of the requesting agent A (step 2). The agent then provides its security requirements and security capabilities to the travel Web service (step 3). The policy engine of the travel Web service then obtains the service workflow from the process ontology and compares it with the security requirements and security capabilities of agent A before providing the requested service (step 4).

6.2.3. Scenario 3: Conference organisation scenario

A scenario presented by Tan and Poslad (2004b) describes an open environment setting where different systems publish their services together with externally-public security configurations and requirements. An agent should be able to discover different services and reason about their security choreography or workflow to support interoperability between disparate services.

A scenario starts with a conference organiser's agent who wishes to organise an event. The agent interacts with services such as a conference hall service, a restaurant service, and a hotel service. The agent discovers these services through directories and then discerns their security choreography. Once the agent has reasoned about the security concerns of the required services, it requests the banking service to make payments to relevant services, and the conference event is organised. This scenario is illustrated in Figure 6.3(a) below.

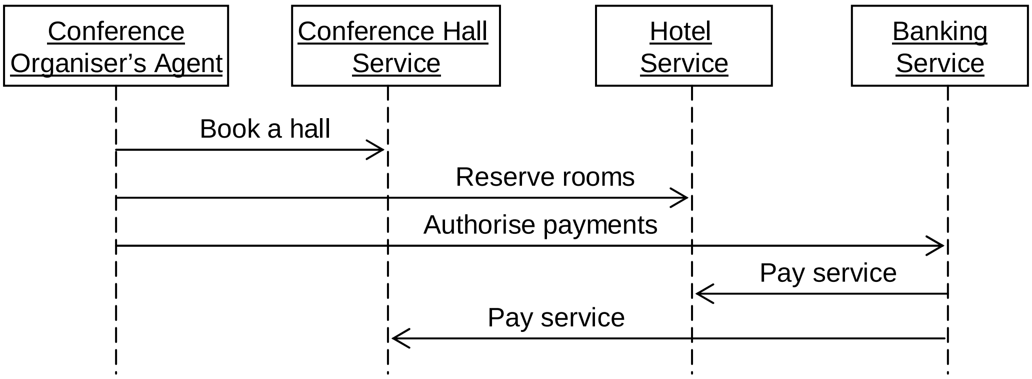


Figure 6.3(a): Conference organisation scenario

The scenario depicts the need for secure service composition and the description of the service workflow to support secure interoperability between disparate services. Reasoning about security concepts and service workflows is needed to facilitate decision-making processes.

The description of service workflow is done by the *process ontology*. The *policy engine* uses information from the process ontology to compose services securely. The service profile of the process ontology may be used to specify inputs, outputs, preconditions, and effects of a service. To compose services from the scenario securely, the process ontology may specify the order in which services should be executed. For instance, from the given scenario the order of service execution could be: book a hall, reserve hotel rooms, make payments, and then confirm the event.

The security concepts described in the *security ontology*, together with the transaction policy and threat-countermeasure policy specified in *policies*, may be used by the *policy engine* to reason about security services to be invoked when making payments to service providers.

The components of the proposed security framework used in this scenario include security ontology, process ontology, transaction policy, threat-countermeasure policy, and the policy engine. The functionalities provided by the proposed security framework include *description of* security concepts, *description of service workflows*, *reasoning of security* *concepts*, and *policy specification*. Figure 6.3(b) illustrates the application of the proposed security framework to the scenario.

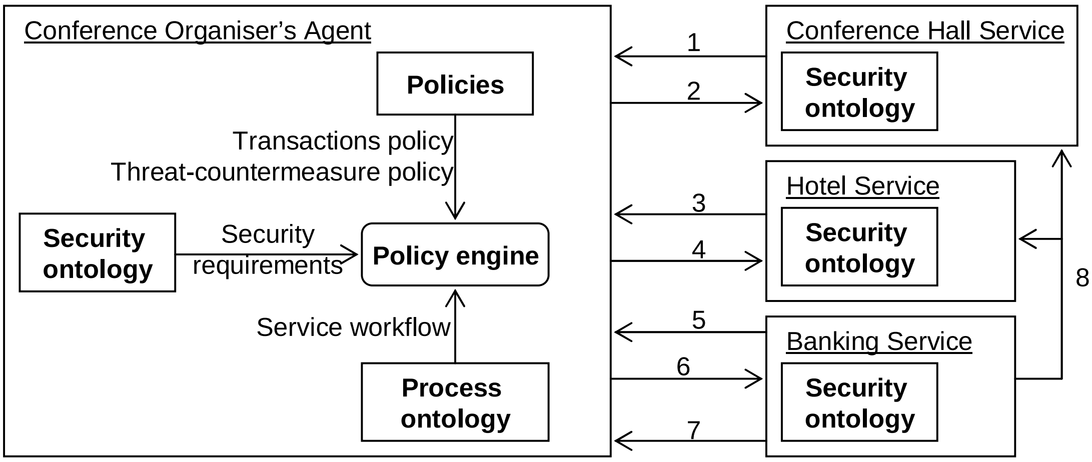


Figure 6.3(b): Secure conference organisation scenario

In a secure conference organisation scenario, the conference organiser's agent retrieves the security capabilities of a conference hall service (step 1). After comparing the security capabilities of a conference hall service with its security requirements by using the policy engine, the conference organiser's agent will then book a conference hall (step 2). Similarly, the conference organiser's agent will retrieve the security capabilities of the hotel service (step 3). After comparing the security capabilities of the hotel service with its security requirements by using the policy engine, the conference organiser's agent will then reserve hotel rooms (step 4). In step 5 the conference organiser's agent retrieves the security capabilities of the banking service. It then uses its policy engine and security policies (transaction policy and threat-countermeasure policy) to reason about the security of the payment transaction. If satisfied with the security capabilities of the banking service, it will then authorise payment for the service (step 6). The banking system will then authenticate the conference organiser's agent (step 7) before it pays for services (step 8). The conference organiser's agent uses its process ontology to compile a service workflow that enforces the order of service execution.

6.2.4. Scenario 4: Meeting scenario

The meeting scenario described by Toninelli et al. (2006) illustrates access control challenges in a dynamic mobile environment. In a meeting scenario participants may wish to give access to their resources to other participants, but the access should be regulated in order to protect the resources from malicious access and misuse. In this scenario, the list of participants is not known a priori, or it can change just before the meeting, or even during the meeting.

The scenario starts with a participant deciding which resources he or she wants to make available to other participants. The resource owner specifies rules or policies to constrain the access to his or her resources. The constraints may include spatial conditions such as requesting the time and location of the requester. Based on the contextual information about the current meeting and the current project discussed in a meeting, access to the requested resources may be granted. Figure 6.4(a) below illustrates the meeting scenario.

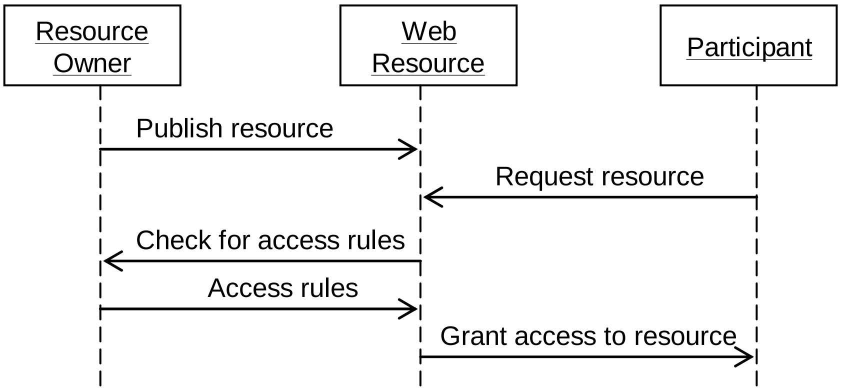


Figure 6.4(a): Meeting scenario

The meeting scenario depicts the need for having access control mechanisms that take into account contextual information such as time and location. The ability to specify security policies dynamically is desirable for cases such as when the meeting continues beyond its original planned end time.

The *identification policy* is used to authenticate the requesting agent and to determine whether the owner of the requesting agent is a participant of a particular meeting.

In the proposed security framework, access control may be specified by using the *access control policy*, in which an entity can describe which entities are allowed to access resources and what type of access is allowed. From the scenario, the access control policy could state that only participants will have read-only access to the minutes of the meeting.

Other contextual information such as time and location may be specified by using the *spatial policy*. The spatial policy could state that access to minutes of the previous meeting will only be granted while the current meeting is on. The *policy engine* reasons about security concepts and policies. Security concepts are specified in the *security ontology*.

Components of the proposed security framework used in this scenario include the security ontology, identification policy, access control policy, spatial policy, and the policy engine. Functionalities provided by the proposed security framework include *policy specification* (access control policy and spatial policy), *reasoning about security concepts* (policy engine), and *description of security concepts* (security ontology). Figure 6.4(b) illustrates the application of the proposed security framework to the scenario.

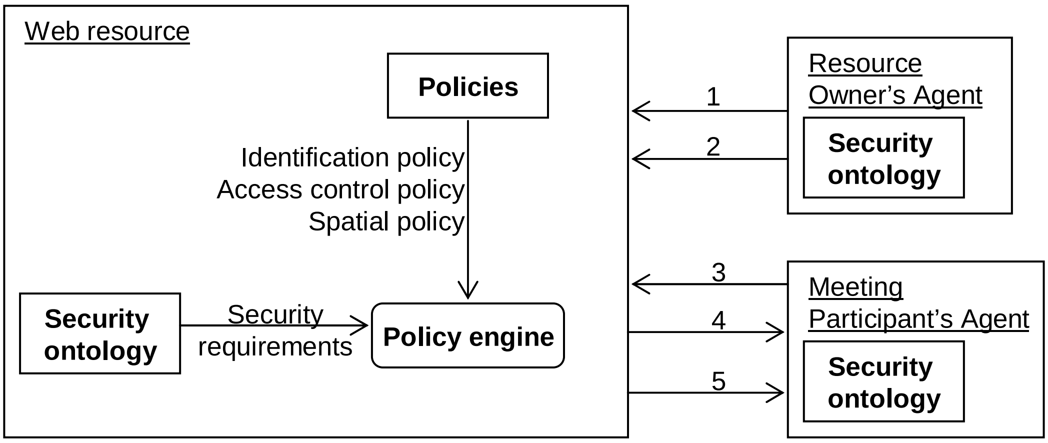


Figure 6.4(b): Secure meeting scenario

In a secure meeting scenario, the resource owner's agent publishes resources that the owner wants to make available to other participants (step 1). The resource owner's agent will then specify security requirements and access control policy for the published resources (step 2). The meeting participants could then use their agents to request access to the published resources (step 3). Upon receiving a request for resource access, the Web resource will use its identification policy to authenticate the requesting agent (step 4). Once the requesting agent has been successfully authenticated, the Web resource will use its access control policy, spatial policies and the policy engine to determine whether to grant access or not. The Web resource will then grant access to the requesting agent or notify the agent of the reasons for denial of access to the requested resource (step 5).

6.2.5. Scenario 5: Hospital information scenario

The hospital information scenario described by Kagal et al. (2003) presents a scenario whereby electronic patient information is redacted according to the requester's credentials, his relationship to the patient and other security requirements set by the hospital website.

This scenario starts when one of the doctors, who is away on leave, is discussing a difficult case with one of his friends, who is also a doctor. His friend makes a suggestion the doctor has not tried yet . The doctor uses his cell phone to retrieve certain information from the hospital's website. The Web server rejects his request because no credentials have been submitted. The doctor then asks his agent to retrieve the information. When the website denies the agent's request, the agent asks for the list of credentials required. The Web server replies that it requires a hospital authorised certificate. The agent is able to understand the credential perfectly and resends the request with the necessary certificate attached. This time the Web server permits the request and the agent returns the information to the requesting doctor. This scenario is illustrated in Figure 6.5(a) below.

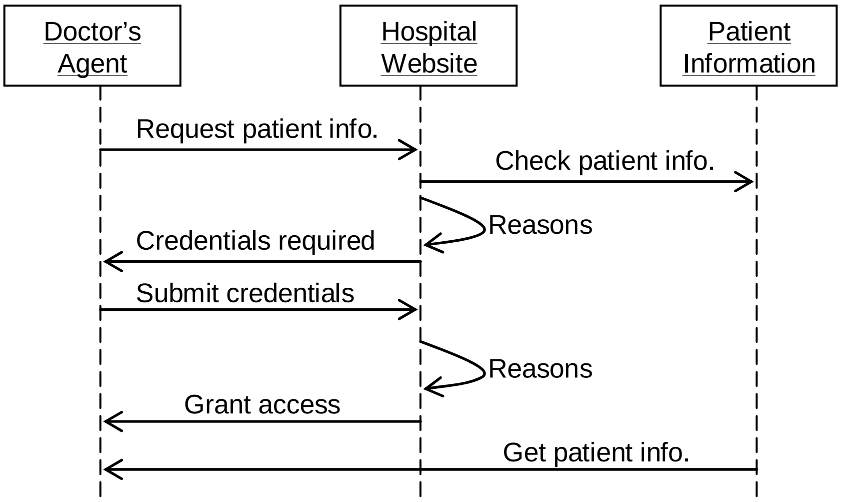


Figure 6.5(a): Hospital information scenario

The hospital information scenario illustrates the need for a dynamic access control mechanism based on the requester's credentials. The issue of privacy in respect of patient information is also paramount in this situation.

The use of semantic languages in specifying security policies allows dynamic adaptation of access control policies. The proposed security framework uses semantic language (OWL and RDF-S) to specify access control policies. The access control policy in this scenario could state that only entities with the authorised hospital certificate could have read-access to patient information.

The *identification policy* could be used by the hospital website to authenticate the requesting doctor's agent. The identification policy could state the credentials that a requesting agent should possess. According to the scenario, the requesting agent must have a hospital authorised certificate.

The *privacy policy* may be used to specify privacy preferences regarding the use of patient information. The privacy policy also specifies with whom the information can be shared, and for how long the information can be retained by the requester. From the scenario, the privacy policy could state that the requester of patient information could only share the information with other doctors who have specific credentials.

The components of the proposed security framework used in this scenario include the security ontology, identification policy, access control policy, privacy policy, and the policy engine. Functionalities provided by the proposed security framework include *description of security concepts* (security ontology), *policy specification* (access control policy and privacy policy) *and reasoning about security concepts* (policy engine). Figure 6.5(b) illustrates the application of the proposed security framework to the scenario.

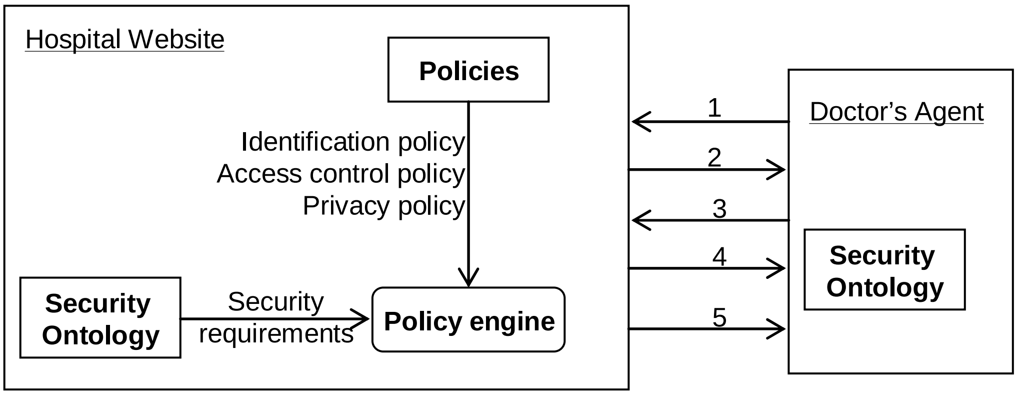


Figure 6.5(b): Secure hospital information scenario

The secure hospital information scenario starts with the doctor's agent sending a request to access patient information from the hospital's website (step 1). On receiving the request to access patient information, the hospital website retrieves its security requirements and sends them to the doctor's agent (step 2). The doctor's agent then will have to send the required credentials to the hospital website (step 3). The hospital website will then use its identification policy to authenticate the doctor's agent (step 4). If the agent is successfully authenticated, the hospital website will use its policy engine, together with access control policy, privacy policy and the credentials submitted by the doctor's agent, to determine whether to grant access or not. The hospital website will then grant access to the patient's information or notify the doctor's agent of the reason for denial of access (step 5).

6.3. CONCLUSION

This chapter presented application of different usage scenarios to the proposed security framework as a proof-of-concept. A variety of usage scenarios have been used to demonstrate the applicability of the proposed security framework to different situations. The scenarios have been selected from various application domains to increase the quality of generalisation.

Issues addressed by the application scenarios include the description of security concept, description of service workflows, policy specification, and reasoning about security concepts and policies. These issues demonstrate the use of different components of the proposed security framework to tackle different security concerns. The components used include security ontology, process ontology, policy, and policy engine. Regarding policy specification, the scenarios demonstrated the use of identification policy, access control policy, transactions policy, privacy policy, threat-countermeasure policy, and spatial policy.

The framework application scenarios helped to demonstrate that the proposed security framework could work in different practical situations. As a proof-of-concept, the scenarios strengthen the merit of the proposed security framework for the Semantic Web. The use of different usage scenarios eliminates misunderstandings about the scope and functionalities of the proposed security framework.

The following chapter concludes this dissertation by presenting the contribution of the study, the summary of findings, and the conclusions and recommendations for future work.

CHAPTER 7: CONCLUSION AND CONTRIBUTION

7.1. INTRODUCTION

In this chapter, the dissertation is concluded by presenting summaries of the research findings and conclusions made by the study. The contributions made by the research, as well as recommendations for further research, are also presented.

In Section 7.2, a summary of research findings is presented, and a summary of conclusions is presented in Section 7.3. The contribution made by the study is presented in Section 7.4, and the recommendations for further research are presented in Section 7.5. Section 7.6 provides the closing remarks.

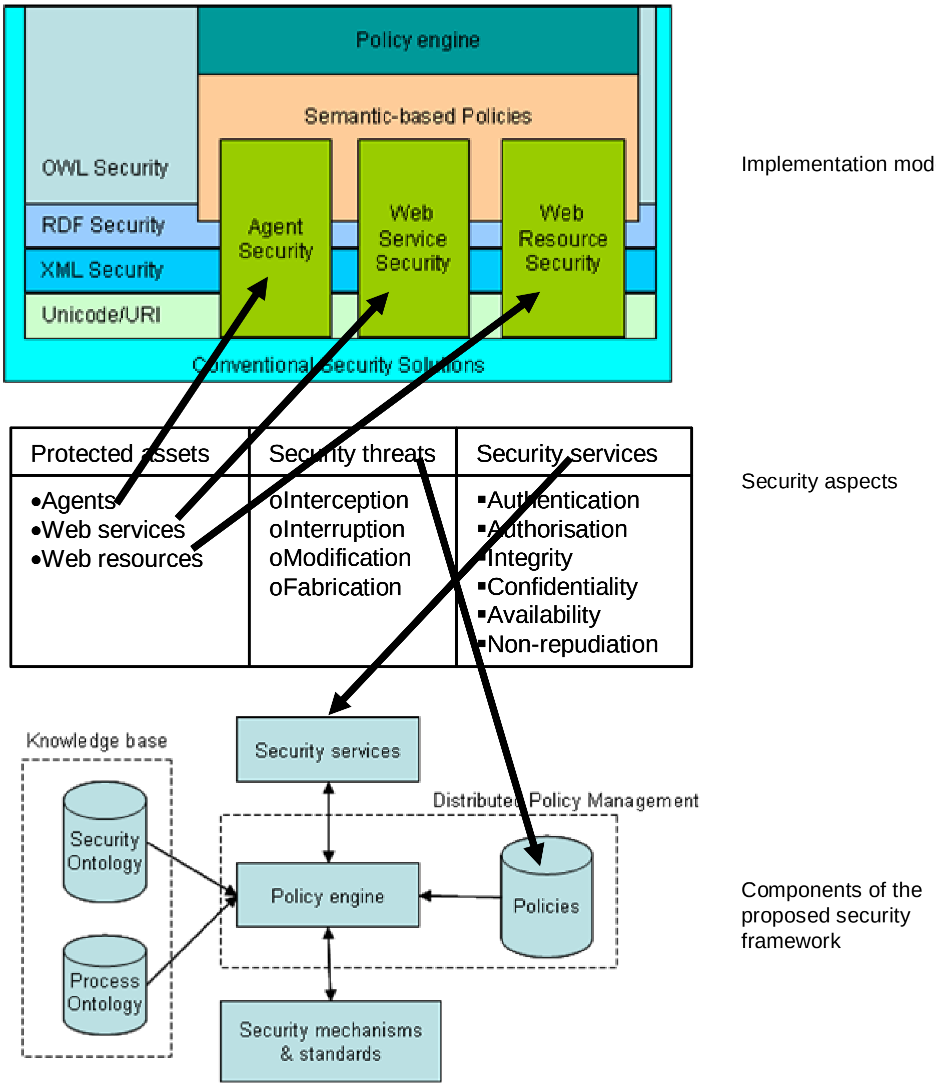
7.2. SUMMARY OF FINDINGS

In the course of answering research questions outlined in Chapter 1, this dissertation presented different findings from each research sub-question. Each chapter of the dissertation, with the exception of Chapter 1 (introduction) and Chapter 3 (research design and methodology), addressed different research questions. This section presents a summary of the research findings extracted from different chapters of the dissertation.

The main objective of this study is to develop a security framework for the Semantic Web. A framework in this study was defined as a brief set of ideas for organising a thought process about a particular type of thing. In the process of developing a security framework for the Semantic Web, several research questions were set and answered.

The first research sub-question to be answered was: *what security aspects are related to the Semantic Web?* Research activities pertaining to this research question were presented in Chapter 2. Security aspects related to the Semantic Web contributed to the extraction of requirements (Section 4.4), the extraction of essential components (Section 4.6), and to the proposed security framework (Section 5.3). Security aspects related to the Semantic Web include the established technologies of the Semantic Web and their functionalities, protected assets in the Semantic Web context, security threats to the Semantic Web environment, and security services (goals) desired for the Semantic Web.

Figure 7.1 below illustrates the input made by the security aspects in building up the proposed security framework for the Semantic Web.



Protected assets	Security threats	Security services
•Agents •Web services •Web resources	oInterception oInterruption oModification oFabrication	▪Authentication ▪Authorisation ▪Integrity ▪Confidentiality ▪Availability ▪Non-repudiation

Figure 7.1: Inputs from security aspects

The established technologies of the Semantic Web and their functionalities, as presented in Section 2.2, include *Unicode and URI* (unique identification), *XML, XML-S,* and *NS* (syntax description languages), *RDF* (metadata data modelling), and *RDF-S* and *OWL* (ontology). These technologies were depicted in the implementation model of the proposed security framework in Figure 5.7 as Unicode/URI, XML security, RDF security, and OWL security.

The protected assets in the Semantic Web environment include the agents, Web services and Web resources. The protected assets were depicted in the implementation model of the proposed security framework in Figure 5.7 as agent security, Web service security, and Web resource security. Security threats associated with the Semantic Web are categorised into *interception, modification, interruption,* and *fabrication*. Security threats were used to adopt security policies depicted in Figure 5.2 as policies. Security services desired to provide a secure environment to the Semantic Web include *authentication, authorisation, integrity, confidentiality, availability,* and *non-repudiation*. Security services contributed to the main components of the proposed security framework as depicted in Figure 5.2. Table 7.1 below summarises security aspects related to the Semantic Web.

Table 7.1: Summary of security aspects for the Semantic Web

Protected assets	Security threats	Security services
• Agents • Web services • Web resources	o Interception o Interruption o Modification o Fabrication	▪ Authentication ▪ Authorisation ▪ Integrity ▪ Confidentiality ▪ Availability ▪ Non-repudiation

The second research sub-question to be answered was: *what are the requirements of a security framework for the Semantic Web?* The question aimed at establishing characteristics that could be used as desiderata of a security framework for the Semantic Web. Research activities pertaining to this research sub-question were presented in Chapter 4 of the dissertation. The research established nine requirements. Table 7.2 below outlines the requirements of a security framework for the Semantic Web.

Table 7.2: Requirements of a security framework for the Semantic Web

No	Requirements
1	Decoupling of security functionalities from core service functionalities
2	Layered security support
3	Flexible, dynamic and adaptive
4	Semantically rich
5	Simple enough to automate
6	Impervious to common network problems
7	Implementable on current Semantic Web technologies
8	Provides protection to all Semantic Web entities
9	Provides a complete set of security services

The third research sub-question to be answered was: *what are the components that we can use from existing security frameworks?* The question aimed at studying the existing security frameworks to determine components applicable to the Semantic Web context. The study of existing security frameworks was presented in Chapter 2. According to existing theory, there exist security frameworks that have some of the essential components of a security framework for the Semantic Web. The essential components of a security framework were extracted from the literature and presented in Chapter 4. Security frameworks with essential components are adapted to satisfy the requirements of a security framework for the Semantic Web. Table 7.3 below summarises the essential components of a security framework.

Table 7.3: Essential components of a security framework

Component	Description
Knowledge base	Stores security-related facts and rules
Security services	Allows specifications of security goals
Policies	Defines sets of rules, policies and constraints to address security threats
Security mechanisms	Describes security standards and specifications that provide particular security services
Reasoning engine	Interprets and reasons about policies and security concepts in order to enforce a security service by using a particular security mechanism

The fourth research sub-question to be answered was: *what are the components of a security framework for the Semantic Web?* The question aimed at identifying components that are needed to compile a security framework for the Semantic Web. Activities relating to this question were presented in Chapter 5. The research established four main components of a security framework for the Semantic Web. The components of the proposed security framework are the result of the adaptation of the essential components to the requirements of a security framework for the Semantic Web as discussed in Section 5.2.

The main components established from the proposed security framework for the Semantic Web include the knowledge base, distributed policy management, security services, and security mechanisms and standards. The functionalities of these components include description of security concepts, reasoning, policy specification, specification of security services, and enforcement of security goals. Table 7.4 below summarises the components of a security framework for the Semantic Web.

Table 7.4: Components of a security framework for the Semantic Web

Main component	**Sub component**
Knowledge base	Security ontology
	Process ontology
Distributed policy management	Policies
	Policy engine
Security services	
Security mechanisms and standards	

The functionalities of the components are summarised on Table 7.5 below.

Table 7.5: Components functionalities

Component	Functionality
Security ontology	Description of security concepts
Process ontology	Description of service workflows
Policies	Policy specification
Policy engine	Reasoning of security concepts and policies
Security services	Specification of security services
Security mechanisms and standards	Specification of security mechanisms and standards

The research findings summarised above provide a way of drawing useful and valid conclusions that are summarised in the next section.

7.3. SUMMARY OF CONCLUSIONS

From the title of this document it is evident that the main objective of this study is to develop a security framework for the Semantic Web. The move towards a security framework for the Semantic Web is motivated by the security challenges discussed in Chapter 2 as well as the non-existence of a security framework that satisfies the requirements of a security framework for the Semantic Web outlined in Chapter 4.

From the research findings summarised in Section 7.2 above, the main research question can be answered at this point. The main research question of this dissertation was: *how can a security framework for the Semantic Web be constructed?* This study came to the following conclusions based on the research findings and analysis presented in Chapters 4, 5, and 6.

It was established in Chapter 2 that in the Semantic Web context entities such as agents, Web resources, and Web services are the assets that need protection. The security framework for the Semantic Web therefore provides different security services to these entities.

The Semantic Web entities (agents, Web resources, and Web services) are vulnerable to various security threats such as interceptions, interruptions, modifications, and fabrications. The security framework for the Semantic Web should therefore provide security services to counter the abovementioned security threats. Security services desired for the Semantic Web include authentication, authorisation, confidentiality, integrity, availability, and non-repudiation. This conclusion is a result of the research findings based on the first research sub-question i.e. *what are the security aspects related to the semantic Web?*

A security framework for the Semantic Web provides several security functionalities in the context of the Semantic Web. The proposed security framework explained in Section 5.3 is the result of the research findings based on the four research sub-questions. In other words, the compiled security framework for the Semantic Web is the result of security aspects (sub-question 1), the requirements of a security framework (sub-question 2), the essential components of a security framework (sub-question 3), and the components of a security framework for the Semantic Web (sub-question 4).

A security framework for the Semantic Web consists of a knowledge base, a policy engine, policies, security services, and security mechanisms and standards. The knowledge base stores facts and constraints concerning different security concepts and therefore supports the description of security concepts including security requirements, security capabilities, and service workflows. In the knowledge base ontologies are used to facilitate processing, sharing and reuse of knowledge between Web entities.

Policies are used to define rules and constraints relating to the operation of entities; hence the proposed security framework supports policy specification to constrain and regulate entities' behaviours. Policies allow automation and reasoning about system behaviours.

The policy engine interprets and reasons about policies and facts from the knowledge base in order to make decisions about applicable security services and security mechanisms in a particular situation. The proposed security framework therefore supports reasoning about security policies and security concepts.

Different Semantic Web applications have different security goals as discussed in Section 2.3. Security goals such as confidentiality, integrity, and so on are specified as security services. The proposed security framework therefore allows the specification of security services applicable for a particular application.

To enforce a particular security service (e.g. confidentiality), a particular security mechanism (e.g. SSL) or groups of security mechanisms are used. The proposed security framework allows the specification of security mechanisms as instances of security concepts and security policies. Security mechanisms include security standards such as XMLEnc, PKI, etc.

Security functionalities of the proposed security framework are illustrated in Chapter 6 by using application scenarios as a proof-of-concept. The application scenarios not only explain how the framework works, but also assist in generalising the applicability of the proposed security framework.

7.4. RESEARCH CONTRIBUTIONS

This section describes the ‘new knowledge’ that the research and its conclusions add to the existing body of knowledge. It also presents the theoretical implications of the contribution.

As stated in Section 5.3, three different domains, namely, the Semantic Web, information systems security, and Web services, have provided input to this research. Contributions from the Semantic Web are associated with agents, Semantic Web technologies, and Web resources. Web services‘ contributions are associated with secure discovery and composition of Web services as well as service workflow. Contributions from information systems security are associated with security threats, countermeasures, security policies, and security frameworks.

The first research contribution is the identification of security aspects of the Semantic Web. The research has contributed by analysing established Semantic Web technologies and their functionalities in relation to the layers of the Semantic Web architecture. Clarifying the functionalities of the Semantic Web technologies helps in determining the location of security functionalities in the layered architecture of the Semantic Web. Identifying security threats to each Semantic Web entity is useful in determining security services to protect the Semantic Web. Categorisation of security threats to the Semantic Web and their relation to security services is another contribution in respect of security aspects of the Semantic Web.

The second research contribution is the extraction of the requirements of a security framework for the Semantic Web as presented in Section 4.4. The requirements are useful in determining whether a particular security framework meets the security needs of a particular system. The requirements were used to evaluate existing security frameworks (Section 4.5) and in the build up of the proposed security framework (Section 5.2).

The third research contribution is the identification of essential components of a security framework as presented in Section 4.6. Essential components of a security framework were extracted from existing security frameworks and were used to compile a security framework for the Semantic Web.

The fourth contribution is the establishment of the components of a security framework for the Semantic Web as presented in Section 5.3.

These components are the adaptation of the essential components to the requirements of a security framework for the Semantic Web.

The last contribution is the actual compilation of the proposed security framework for the Semantic Web and its evaluation against the requirements of a security framework for the Semantic Web. It has been argued in Section 5.4 that the proposed security framework satisfies all the requirements of a security framework.

7.5. RECOMMENDATIONS FOR FURTHER RESEARCH

This section presents recommendations for further research based on the findings of the dissertation and the conclusions drawn therein. The study makes recommendations about improving the framework, further validation of the framework, investigating new security frameworks, investigating more requirements, and implementing the framework.

Implementing the framework

This study involved the theoretical development of a security framework for the Semantic Web. Practical issues associated with the framework can only be dealt with once the framework or its prototype is implemented. Implementing the framework will help in deployment of the framework and evaluation of implementation issues.

Improving the framework

The proposed security framework for the Semantic Web utilises the *established technologies* of the Semantic Web outlined in Section 2.2. The framework can be improved by including the *emerging functionalities* of the Semantic Web such as rules, logic, proof, and trust management. The framework can also be improved by considering performance issues and design issues associated with adaptability, expandability, and durability.

Further validation of the framework

The proposed security framework for the Semantic Web involved the adaptation and integration of different components from different security frameworks. The adaptation and integration of the components used in the framework have been done conceptually and without the criteria to validate the outcome. Further validation will improve internal accuracy of the framework.

Investigating new security frameworks

Security for the Semantic Web is currently an active research activity, hence the need to investigate new security frameworks that are currently being developed.

Investigating more requirements

The requirements of a security framework for the Semantic Web were extracted from the existing security frameworks discussed in the literature review. Since the Semantic Web is still in the development stage and most Semantic Web applications are at research level, there is still room to extract new requirements of a security framework for the Semantic Web.

7.6. CLOSING REMARKS

This study developed a security framework for the Semantic Web based on existing Semantic Web technologies. The development of the security framework is justified by the security challenges associated with the dynamicity, openness, heterogeneity, autonomy and distributed nature of the Semantic Web. The study contributed in the area of the Semantic Web, Web services, and information systems security.

REFERENCES

ADAMS, C. & BOEYEN, S. (2002) UDDI and WSDL extensions for Web services: A security framework. *ACM workshop on XML security.* Fairfax, VA, USA, ACM.

AGARWAL, S. & SPRICK, B. (2004) Access Control for Semantic Web Services. *IEEE International Conference on Web Services.* Los Alamitos, CA, USA, IEEE Computer Society.

ALTER, S. (1996) *Information Systems: a management perspective,* Menlo Park, CA, The Benjamin/Cummings Publication Company, Inc.

ANKOLEKAR, A., BURSTEN, M., HOBBS, J. R., LASSILA, O., MARTIN, D. L., DREW MCDERMOTT, S. A. M., NARAYANAN, S., PAULOCCI, M., PAYNE, T. R. & SYCARA, K. (2001) DAML-S: Web Service Description for the Semantic Web. *The First Semantic Web Working Symposium.*

ASHRI, R., PAYNE, T., MARVIN, D., SURRIDGE, M. & TAYLOR, S. (2004) Towards a Semantic Web Security Infrastructure. *AAAI Spring Symposium on Semantic Web Services.*

AVISON, D. & FITZGERALD, G. (2006) *Information Systems Development,* Berkshire, UK, McGraw-Hill Education.

BARTEL, M., BOYER, J., FOX, B., LAMACCHIA, B. & SIMON, E. (2002) XML Signature: Syntax and processing. W3C website. http://www.w3.org/TR/xmldsig-core/

BASS, L., CLEMENTS, P. & KAZMAN, R. (2003) *Software Architecture in Practice,* Boston, MA, Addison-Wesley. Pearson Education, Inc.

BECHHOFER, S., VAN HARMELEN, F., HENDLER, J., HORROCKS, I., MCGUINNESS, D. L., PATEL-SCHNEIDER, P. F. & STEIN, L. A. (2004) OWL Web Ontology Language Reference. W3C Website. http://www.w3.org/TR/2004/REC-owl-ref-20040210/

BERNERS-LEE, T. (1997) Axioms of Web Architecture: Metadata Architecture. W3C Website. http://www.w3.org/DesignIssues/Metadata.html

BERNERS-LEE, T. (2000) Semantic Web - XML2000. W3C Website. http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html.

BERNERS-LEE, T. (2003) Standards, Semantics and Survival. *SIIA Upgrade***,** 6-10.

BERNERS-LEE, T. (2005) WWW 2005 Keynote. W3C Website. http://www.w3.org/2005/Talks/0511-keynote-tbl/

BERNERS-LEE, T. (2006) Artificial Intelligence and the Semantic Web: AAAI2006 Keynote. W3C Website. http://www.w3.org/2006/Talks/0718-aaai-tbl/Overview.html

BERNERS-LEE, T., HENDLER, J. & LASSILA, O. (2001) The Semantic Web. *Scientific American,* 285 (5)**,** 34-44.

BERNERS-LEE, T. & MASINTER, L. (1994) IETF RFC1738 - Uniform Resource Locators (URL): Generic Syntax. http://www.rfc-archive.org/getrfc.php?rfc=1738

BHARGAVAN, K., FOURNET, C. & GORDON, A. D. (2004) Verifying Policy-Based Security for Web Services. *CCS'04.* Washington, DC, USA, ACM.

BRAY, T., HOLLANDER, D. & LAYMAN, A. (1999) Namespaces in XML Recommendation. W3C Website. http://www.w3.org/TR/REC-xml-names/

BRAY, T., HOLLANDER, D., LAYMAN, A. & TOBIN, R. (2006) Namespaces in XML 1.0 (Second Edition). W3C Website. http://www.w3.org/TR/REC-xml-names/

BRAY, T., PAOLI, J., SPERBERG-MCQUEEN, C. M., MALER, E. & YERGEAU, F. (2004) Extensible Markup Language (XML) 1.0 (Third Edition). W3C Website. http://www.w3.org/TR/REC-xml/

BRAY, T., PAOLI, J., SPERBERG-MCQUEEN, C. M., MALER, E. & YERGEAU, F. (2006) Extensible Markup Language (XML) 1.0 (Fourth Edition). W3C Website. http://www.w3.org/TR/2006/REC-xml-20060816/

BRICKLEY, D. & GUHA, R. (2003) RDF Vocabulary Description Language 1.0: RDF Schema. W3C Website. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

BUSSLER, C., FENSEL, D. & MAEDCHE, A. (2002) A Conceptual Architecture for Semantic Web Enabled Web Services. *ACM SIGMOD, SPECIAL ISSUE: Special section on semantic web and data management,* 31 (4)**,** 24-29.

CARMINATI, B., FERRARI, E. & THURAISINGHAM, B. (2004) Using RDF for policy specification and enforcement. *15th International Workshop on Database and Expert Systems Applications.* Los Alamitos, CA, USA, IEEE Computer Society.

CLAESSENS, J., PRENEEL, B. & VANDEWALLE, J. (2001) Combining World Wide Web and Wireless Security. *Network Security***,** 153-172.

CLAESSENS, J., PRENEEL, B. & VANDEWALLE, J. (2003) (How) Can Mobile Agents Do Secure Electronic Transactions on Untrusted Hosts? A Survey of the Security Issues and the Current Solutions. *ACM Transactions on Internet Technology,* 3**,** 28-48.

COVER, R. (2006) SAML: Technology report. Coverpages Website. http://xml.coverpages.org/saml.html

DECKER, S., MELNIK, S., VAN HARMELEN, F., FENSEL, D., KLEIN, M., BROEKSTRA, J., ERDMANN, M. & HORROCKS, I. (2000a) The Semantic Web: The Roles of XML and RDF. *IEEE Internet Computing,* 4**,** 63.

DECKER, S., MITRA, P. & MELNIK, S. (2000b) Framework for the Semantic Web: An RDF Tutorial. *IEEE Internet Computing,* 4 (6)**,** 68-73.

DEMCHENKO, Y., GOMMANS, L., DE LAAT, C., OUDENAARDE, B., TOKMAKOFF, A., SNIJDERS, M. & VAN BUUREN, R. (2005) Security Architecture for Open Collaborative Environment. *Advances in Grid Computing. European Grid Conference 2005.* Berlin, Germany, Springer-Verlag.

DENKER, G., KAGAL, L., FININ, T., PAULOCCI, M. & SYCARA, K. (2003) Security for DAML Web Services: Annotation and Matchmaking. *2nd International Semantic Web Conference. ISWC'03.* Springer-Verlag.

DENKER, G., NGUYEN, S. & TON, A. (2004) OWL-S Semantics of Security Web Services: a Case Study. *Semantic Web: Research and Applications. First European Semantic Web Symposium.* Berlin, Germany, Springer-Verlag.

DETSCH, A., GASPARY, L. P., BARCELLOS, M. P. & CAVALHEIRO, G. G. H. (2004) Towards a Flexible Security Framework for Peer-to-Peer based Grid Computing. *Middleware for Grid Computing***,** 52-56.

EUZENAT, J. & NAPOLI, A. (2003) The Semantic Web: Year One. *IEEE Intelligent Systems,* 1094-7167 (3).

FARKAS, C. & HUHNS, M. N. (2002) Making Agents Secure on the Semantic Web. *IEEE Internet Computing***,** 76-79.

FENSEL, D. (2000) The semantic Web and its languages. *IEEE Intelligent Systems,* 15 (6)**,** 67-73.

FENSEL, D. (2002) Language Standardization for the Semantic Web: The Long Way from OIL to OWL. *Distributed Communities on the Web: 4th International Workshop, DCW 2002.* Sydney, Australia.

FININ, T. & JOSHI, A. (2002) Agents, Trust, and Information Access on the Semantic Web. *SIGMOD Record,* 31 (4)**,** 30-35.

FORD, W., HALLAM-BAKER, P., FOX, B., DILLAWAY, B., LAMACCHIA, B., EPSTEIN, J. & LAPP, J. (2001) XML Key Management Specification (XKMS). W3C Website. http://www.w3.org/TR/2001/NOTE-xkms-20010330/

GASSER, M., GOLDSTEIN, A., KAUFMAN, C. & LAMPSON, B. (1989) The Digital Distributed System Security Architecture. *National Computer Security Conference.*

GERBER, A. (2007) Towards a Comprehensive and Functional Layered Architecture for the Semantic Web. Pretoria, UNIVERSITY OF SOUTH AFRICA.

GERBER, A., BARNARD, A. & VAN DER MERWE, A. (2006) A Semantic Web Status Model. *Proceedings of the Ninth World Conference on Integrated Design & Process Technology.* San Diego, California, IEEE.

GONZALEZ, A. & DANKEL, D. D. (1993) *The Engineering of Knowledge-Based Systems: Theory and Practice,* New Jersey, Prentice Hall.

GRAU, B. C. (2004) A Possible Simplification of the Semantic Web Architecture. *Proceedings of the 13th international conference on World Wide Web, WWW '04.* New York, NJ, USA, ACM Press.

HEFLIN, J. (2004) OWL Web Ontology Language, Use Cases and Requirements. W3C Web site. http://www.w3.org/TR/2004/REC-webont-req-20040210/

HENDLER, J. (2001) Agents and the Semantic Web. *IEEE Intelligent Systems,* 16.

HITZLER, P., ANGELE, J., MOTIK, B. & STUDER, R. (2005) Bridging the Paradigm Gap with Rules for OWL. *In Proceedings of the W3C Workshop on Rule Languages for Interoperability.*

HOFSTEE, E. (2006) *Constructing a Good Dissertation: A practical guide to finishing a Masters, MBA or PhD on schedule,* Johannesburg, South Africa, EPE.

HORROCKS, I., PARSIA, B., PATEL-SCHNEIDER, P. & HENDLER, J. (2005) Semantic Web Architecture: Stack or Two Towers? *Lecture Notes in Computer Science: Principles and Practice of Semantic Web Reasoning: Third International Workshop.*

HORROCKS, I. & PATEL-SCHNEIDER, P. F. (2003) Three theses of representation in the semantic web. *Proceedings of the 12th international conference on World Wide Web. WWW '03.* New York, NY, USA, ACM Press.

HORROCKS, I., PATEL-SCHNEIDER, P. F., BOLEY, H., TABET, S., GROSOF, B. & DEAN, M. (2004) SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Website. http://www.w3.org/Submission/SWRL/

HOWARD, R. & KERSCHBERG, L. (2004) A Framework for Dynamic Semantic Web Services Management. *International Journal of Cooperative Information Systems,* 13 (4)**,** 441-485.

HUANG, D. (2005) Semantic Policy-based Security Framework for Business Processes. *Semantic Web and Policy Workshop.* Galway, Ireland.

HUANG, D. (2006) Semantic Description of Web Services Security Constraints.

IETF (2003) Overview of Internet Standards (STD) in Numeric Order. IETF Website. http://www.rfc-archive.org/standards.php

IETF (2006) Internet Engineering Task Force. Web document. http://www.ietf.org

IMAMURA, T., DILLAWAY, B. & SIMON, E. (2002) XML Encryption Syntax and Processing. W3C Website. http://www.w3.org/TR/xmlenc-core

ISO7498-2 (1988) Security Architecture. International Standard Organisation.

JANSEN, W. A. (2000) Countermeasures for Mobile Agent Security. *Computer Communications, Special Issue on advanced security techniques for network protection,* 23 (17).

JIANG, G., CHUNG, W. & CYBENKO, G. (2003) Dynamic Integration of Distributed Semantic Services. *International Conference on Integration of Knowledge Intensive Multi-Agent Systems, KIMAS'03: Modelling, Exploration, and Engineering.* Boston, MA, USA, IEEE Computer Society.

JOSHI, A., FININ, T. & YESHA, Y. (2002) Me-Services: A Framework for Secure & Personalised Discovery, Composition and Management of Services in Pervasive Environments. *Web Services, E-Business, and the Semantic Web. WES'02.* Berlin, Germany, Springer-Verlag.

KAGAL, L., BERNERS-LEE, T., CONNOLY, D. & WEITZNER, D. (2006) Self-describing Delegation Networks for the Web. *Seventh IEEE International Workshop on Policies for Distributed Systems and Networks.* Los Alamitos, CA, USA, IEEE Computer Society.

KAGAL, L., FININ, T. & JOSHI, A. (2003) A Policy-Based Approach to Security for the Semantic Web. *2nd International Semantic Web Conference. ISWC'03.* Springer-Verlag.

KARNIK, N. (2000) Security in Mobile Agent Systems. University of Minnesota.

KEMMERER, R. A. (1998) Security Issues in Distributed Software. Santa Barbara, University of California.

KLYNE, G. (2002) Framework for Security and Trust Standards. SWAD Europe project. http://www.w3.org/2001/sw/Europe/

KNIGHT, W. (2004) Goin' phishing? *Infosecurity Today*, Elsevier.

LEE, J. K., UPADHYAYA, S. J., RAO, H. R. & SHARMAN, R. (2005) Secure Knowledge Management and the Semantic Web. *Communications of the ACM,* 48 (12)**,** 48-54.

LI, C. & PAHL, C. (2003) Security in the Web Services Framework. Dublin, Dublin City University.

MCGUINNESS, D. L. (2004) Question Answering on the Semantic Web. *IEEE Intelligent Systems,* 19 (1)**,** 82-85.

MCGUINNESS, D.L., FIKES, R., HENDLER, J. & STEIN, L.A. (2002) DAML+OIL: An Ontology Language for the Semantic Web. *IEEE Intelligent Systems*, 72–80.

MCGUINNESS, D. L. & VAN HARMELEN, F. (2004) OWL Web Ontology Language Overview. W3C Website. http://www.w3.org/TR/2004/REC-owl-features-20040210/

MOUTON, J. (2005) *How to succeed in your Master's & Doctoral Studies: A South African guide and resource book,* Pretoria, South Africa, Van Schaik Publishers.

OLIVIER, M. S. (2004) *Information Technology Research: A practical guide for Computer Science and Informatics,* Pretoria, South Africa, Van Schaik Publishers.

OMG (2000) Agent Technology Green Paper. Object Management Group. http://www.omg.org/

PALLICKARA, S. & FOX, G. (2003) A Security Framework for Distributed Brokering Systems. *Proceedings of ACM/IFIP/USENIX International Middleware Conference Middleware.*

PALMER, S. B. (2001) The Semantic Web: An Introduction. Infomesh Website. http://infomesh.net/2001/swintro/

PARK, J. S. (2003) Towards Secure Collaboration on the Semantic Web.

PATEL-SCHNEIDER, P. F. & FENSEL, D. (2002) Layering the Semantic Web: Problems and Directions. *Proceedings of The Semantic Web - ISWC 2002: First International Semantic Web Conference.* Sardinia, Italy, Springer-Verlag.

PFLEEGER, C. P. & PFLEEGER, S. L. (2003) *Security in Computing,* Upper Saddle River, NJ, Prentice Hall.

POD (2004). Pocket Oxford Dictionay, Oxford University Press, Oxford.

PRESSMAN, R. S. & INCE, D. (2000) *Software Engineering: A Practitioner's Approach. European Adaptation,* Berkshire, UK, McGraw-Hill Publishing Company.

QIN, L. & ATLURI, V. (2003) Concept-Level Access Control for the Semantic Web. *ACM Workshop on XML Security.* Fairfax, VA, USA, ACM.

SHIELDS, B., MOLLOY, O., LYONS, G. & DUGGAN, J. (2006) Using Semantic Rules to Determine Access Control for Web Services. *WWW 2006.* Edinburgh, Scotland, ACM. 1595933329/06/0005.

SINGH, R. & SALAM, A. F. (2006) Semantic Information Assurance for Secure Distributed Knowledge Management: A Business Process Perspective. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans,* 36 (3)**,** 472-486.

SMITH, M. K., WELTY, C. & MCGUINNESS, D. L. (2004) OWL Web Ontology Language Guide. W3C Website. http://www.w3.org/TR/2004/REC-owl-guide-20040210/

TAN, J. J. & POSLAD, S. (2004a) Dynamic security reconfiguration for the semantic web. *Engineering Applications of Artificial Intelligence,* 17 (7)**,** 783-797.

TAN, J. J. & POSLAD, S. (2004b) A Profile Based Security Model for the Semantic Web. *ECOWS'04.* Berlin, Germany, Springer-Verlag.

THURAISINGHAM, B. (2002) Building secure survivable semantic webs. *14th IEEE International Conference on Tools with Artificial Intelligence. ICTAI'02.* IEEE Computer Society.

THURAISINGHAM, B. (2003) Security Issues for the Semantic Web. *27th Annual International Computer Software and Applications Conference. COMPSAC'03.* IEEE Computer Society.

THURAISINGHAM, B. (2005) Directions for Security and Privacy for Semantic E-Business Applications. *Communications of the ACM,* 48 (12)**,** 71-3.

TONINELLI, A., MONTANARI, R., KAGAL, L. & LASSILA, O. (2006) A Semantic Context-Aware Access Control Framework for Secure

Collaborations in Pervasive Computing Environments. *5th International Semantic Web Conference, ISWC 2006.* Berlin, Germany, Springer-Verlag.

TURNER, A., DOGAC, A. & TOROSLU, I. H. (2005) A Semantic-Based User Privacy Protection Framework for Web Services. *Intelligent Techniques for Web Personalisation. ITWP'03.* Berlin, Germany, Springer-Verlag.

UNICODE (2004) The Unicode Consortium. Unicode Website. http://www.unicode.org/

UNICODE (2006) The Unicode Standard. Unicode Website. http://www.unicode.org/standard/versions

USCHOLD, M. (2003) Where Are the Semantics in the Semantic Web? *AI Magazine,* 24 (3)**,** 25–36.

USZOK, A., BRADSHAW, J. M. & JEFFERS, R. (2004) KAoS: A Policy and Domain Services Framework for Grid Computing and Semantic Web Services. *Trust Management. Second International Conference, iTrust '04.* Berlin, Germany, Springer-Verlag.

USZOK, A., BRADSHAW, J. M., JOHNSON, M., JEFFERS, R., TATE, A., DALTON, J. & AITKEN, S. (2004) KAoS Policy Management for Semantic Web Services. *IEEE Intelligent Systems***,** 32-41.

VENTUNEAC, M., COFFEY, T. & SALOMIE, I. (2003) A Policy-Based Security Framework for Web-Enabled Applications. *2nd International Semantic Web Conference. ISWC'03.*

VUONG, S. T. & FU, P. (2002) A Security Architecture and Design for Mobile Intelligent Agent Systems. Vancouver, The University of British Columbia, Canada.

W3C (1999) Resource Description Framework (RDF) Model and Syntax Specification. W3C Website. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

W3C (2001a) XML Schema Part 1: Structures - W3C Recommendation. W3C Website. http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

W3C (2001b) XML Schema Part 2: Datatypes - W3C Recommendation. W3C Website. http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

W3C (2004a) RDF Primer Recommendations. W3C Website. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

W3C (2004b) RDF Primer. W3C Website. http://www.w3.org/TR/rdf-primer/

W3C (2004c) RDF Vocabulary Description Language 1.0: RDF Schema. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

W3C (2004d) World Wide Web Consortium Issues RDF and OWL Recommendations. W3C Website. http://www.w3.org/2004/01/sws-pressrelease

W3CRULE (2005) W3C Rule Working Group. W3C Website. http://www.w3.org/2005/rules/wg

WALLACE, R. (2002) *MCSE Training Kit: Microsoft Windows XP Professional,* Redmond, Washington, USA, Microsoft Press.

WHITTEN, J. L., BENTLEY, L. D. & BARLOW, V. M. (1994) *Systems Analysis and Design Methods,* Boston, MA, USA, IRWIN.

XU, B., LI, Y., LU, J. & KANG, D. (2006) Secure OWL Query. *Computer Science-ICCS. 6th International Conference.* Berlin, Germany, Springer-Verlag.

YAGUE, M. I., MANA, A., LOPEZ, J. & TROYA, T. (2003) Applying the Semantic Web Layers to Access Control. *14th International Workshop on Database and Expert Systems Applications (DEXA'03).*