Creating a Directory protocol specification for networked services

Master's thesis, April 8, 2010

Ruben van Eijnatten

Creating a Directory protocol specification for networked services

THESIS

submitted in partial fulfilment of

the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

TRACK INFORMATION ARCHITECTURE

by

Ruben van Eijnatten

born in Huizen, The Netherlands

Web Information Systems Group
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, The Netherlands
http://eemcs.tudelft.nl

Innopay

Schiphol-Airport, The Netherlands
www.innopay.com

Creating a Directory protocol specification for networked services

Author: Ruben van Eijnatten

Student id: 1149792

Email: r.vaneijnatten@student.tudelft.nl

Abstract

Technological and economical developments create new market opportunities for networked services. Many of these opportunities are addressed, however for many new developments the infrastructure is reinvented while these services have similar infrastructural requirements. This reinvention of infrastructures is a costly activity; hence there exists a need for standardization and reuse.

Innopay has recognized this need and set out to create the SIX standards (1) that must provide standardized building blocks for schemed networked service infrastructures; thus reducing the costs of infrastructure development and allowing the service developers to direct their focus to innovation in the application domain.

Two important building blocks for such an infrastructure are a secure and reliable messaging interface and a directory protocol. For service providers in the network to securely and reliably exchange messages, a method is needed to exchange information for identification, addressing and authentication.

The research documented in this report provides a solution for this data distribution problem. The primary result includes a specification based on ebXML RS/RIM (2) of a protocol for configuration and operation of a directory and the interaction of users with the directory.

Graduation Committee:
prof. dr. ir. G.J. Houben, Faculty EEMCS, TU Delft
ir. B. R. Sodoyer, Faculty EEMCS, TU Delft
Vincent Jansen, Innopay
dr. P. G. Kluit, Faculty EEMCS, TU Delft

Preface

The document before you documents the research conducted as part of the final project of the Computer Science master program variant Information Architecture of the Delft University of Technology in Delft, The Netherlands.

The research was conducted with the support of Innopay, a leading payments consultancy firm located at Schiphol-Airport, The Netherlands from March 2009 until April 2010.

The document describes and discusses the process of creating a directory protocol specification for networked services. The intended reader is the technical decision makers seeking background information on the directory protocol specification. No prior knowledge is assumed other than a general technical background in information technology or computer science. For a full account of the early stages of the process the reader is referred to the preliminary requirements analysis and technology survey (3) for more information.

The research and documentation of it would not have been possible without the valuable help of a number of people which I would like to express my great gratitude to.

First, I would like to thank my supervisors for their guidance throughout the project. I am very thankful for the valuable feedback provided by Bernard Sodoyer and his patience and understanding and also for the many insightful comments provided by Vincent Jansen and the generous sharing of his extensive experience.

Second, I would like to express my gratitude to Innopay, especially Chiel Liezenberg and Douwe Lycklama for providing the opportunity to conduct the project in a challenging and inspiring environment. Innopay is the leading company in electronic transaction service consultancy in Europe and I cannot imagine a better place to have conducted this research at. Also, many thanks to the members of the expert group which I consulted regularly: Vincent Jansen, Chiel Liezenberg, David van den Hengel, Jelle-Frodo Huisman and Leendert Bottelberghs.

Last, but certainly not least, I would like to thank my loving family, close friends and my colleagues at Greetinq, for their undying support, care and understanding during the whole project.

Finally I hope all readers will find this document and its contents helpful.

Ruben van Eijnatten
Delft, the Netherlands
April 8th, 2010

Contents

Table of figures

This table provides an overview of the figures in this document.

PART 1 PRELIMINARIES

This part introduces the research of which this report is the documentation and discusses the background of the research and the context in which the research has taken place.

1 Introduction

This section introduces the research documented in this report. The first section contains the problem statement of which the research goal is derived in section 1.2. The third section discusses the research questions that when answered lead to the accomplishment of the research goals. The approach taken to answer the questions is outlined in 1.5. The outline of the rest of this report is discussed section 1.6.

1.1 Problem statement

The continuing technological and economical developments have created numerous market opportunities for networked services. Online networked services are services that are delivered by two or more parties in cooperation, where the service delivery involves the exchange of information over the internet (online) and in (near) real time. The relevant developments are discussed further in chapter 2.

This type of service has specific infrastructure requirements, such as reliable and secure messaging, that are not fulfilled by traditional internet technology and thus infrastructure designers are required to specify additional technology on top of common internet technology standards. Since there is no appropriate standardized solution, this activity is repeated for each project.

While being involved in the development of a number of national and international online networked service schemes, including the Dutch online payments scheme iDEAL (4) and the online invoicing scheme Standaard Digitale Nota, Innopay has experienced this lack of standardization first hand. As a result Innopay has started to draft the SIX standards. The SIX standards suite is a set of specifications aimed to provide a set of generic infrastructure building blocks for infrastructure developers.

One essential part of this set of specifications is the Secure SOAP interface (5) that prescribes how a secure and reliable point-to-point connection should be set up. The secure interface requires the distribution of information necessary to reliable, secure and real-time communication between all participants.

To solve this information distribution problem a protocol is needed to specify how service providers in the network can obtain and share this information; a directory protocol that solves the discoverability and addressing issues and creates trust and security, while increasing reliability.

1.2 Research goal

The goal of this research is to provide a solution to this information distribution problem for multi-party networked services by selecting or creating a specification of a directory protocol and with that provide networked service infrastructure designers with a standardized building block.

A technology survey conducted prior to this research project has revealed the existence of a number of technologies that can potentially contribute to providing a solution to (part of) the problem. An important part of this research is to assess the extent to which

these technologies fulfill the requirements that where drafted as a result of the requirements analysis conducted prior to this research. (3)

In addition to basing the directory protocol specification on an existing technology specification it is possible to draft a new specification. The requirements prescribe that the specification shall be based on existing standards an specification as much as possible to improve the ease of adoption of the new specification. A challenge is to determine the trade-off between an existing solution with limited functional fit but better scores for ease of adoption; and a tailor-made solution with near-perfect functional fit but low scores on standardization and use (both contributing to ease of adoption).

1.3 Research questions

In able to reach the research goals the following research questions are to be answered in by the research. The main research question is: *How to solve the data distribution problem for multi-party networked services?*

A solution to the data distributions problem is a method for distributing information to all parties that need it; This information is needed by those parties to be able to securely and reliably exchange messages.

Multi-party networked services are services with which two or more parties are involved in the service delivery. Networked services are often found in two-sided market facilitating the interaction between both sides of the market.

In addition to the main research question the following questions are answered in this research:

- *How to assess the suitability of existing specifications?*
- *Which existing specification(s) is most suited to base the new specification on?*
- *How does use of this/these specification(s) compare to creating a new specification?*

Earlier research (3) has revealed the existence of a number of potential (partial) solutions to this problem. As part of this research the suitability of these existing specifications need to be assessed to determine the extent to which these specifications provide a solution to the problem. A suitable method for assessing the existing specification needs to be determined.

- *How to draft a specification?*
- *How to validate the suitability of the specification?*

Once the assessment has resulted in the selection and discard of existing specifications to include in the solution, a new specification has to be drafted that describes the new solution. The method for drafting this specification needs to be determined.

1.4 Research results

The results of this research effort include both a specification of a directory protocol and documentation of the rationale behind the specification.

The directory protocol specification contains a number of normative statements that restrict the implementation and use of technology that implements the directory protocol. It specifies the protocol to use when interacting with a compliant directory service and the behavior of the directory service as a result of this interaction.

The directory protocol specification is included as one of the core specifications of the SIX standards. Together with the SIX:0208 Secure SOAP Interface (5) and the SIX:2909 Real-time online protocol is to be developed, the new specification forms a building block for networked service designers.

It is the intention of Innopay to elevate the SIX specifications to a set of standards that are widely adopted. The rationale behind the specification is also considered a part of the results of this research as it is a crucial part in the standardization process that this specification will enter along with the other SIX specifications.

1.5 Research approach

The following approach is taken for this research. The approach contains four steps that are discussed below: requirements refinement, technology assessment, drafting of the specification and standardization of the specification.

Requirement refinement

The requirements analysis conducted prior to this research (3) serves as a starting point for this research. The set of requirements was compiled for the purpose of conducting a technology survey. The technology survey (3) had to identify potential relevant existing technologies and specifications.

The level of detail used to specify the requirements was sufficient for this purpose; however for the in depth assessment of the existing technology a higher level of detail is necessary. The first step in this research is to increase the level of detail in the requirements specification. From the high-level requirements one or more low-level requirements can be derived.

The purpose of the refined requirements specification is to serve as input for the technology assessment and later for the drafting of the specification. The level of detail for the refinement should be adequate for this purpose.

Technology assessment

The second step in this research approach is an assessment of existing technologies, products and standards that are identified as potential (partial) solutions to the problem. The assessment should be concluded with selection and discard of technologies to incorporate in the new specification. Potentially none of the technologies is selected and it is concluded that a completely new specification needs to be drafted.

The input for the assessment consists of the results of the technology survey on one hand and the results of the requirements analysis and refinement on the other hand. The

technology survey results provide the technologies to assess. The requirements analysis results provide the assessment criteria.

The assessment methodology takes an iterative approach and consists of three iterations in which the level of detail of the criteria is increase for each iteration. The assessment methodology is presented in more depth in section 5.1.

Specification drafting
At this point the existing technologies to be used in the new specification are determined. If more than one existing technology is selected an additional document should specify how to combine these technologies and how to use the combination in the specific context. If a single technology is selected it is likely to need additional specification on how to extend and use this technology for the specific context. In case no technology is selected at all a solution needs to be specified from scratch.

The specification to be drafted contains two parts. The first part specifies how the solution should be set up. The second part specifies how the solution should be used. The specification should provide infrastructure developers with a ready-to-use building block for infrastructures in the networked services context.

Standardization
A final step of this research is to provide recommendation on the steps to be taken to elevate the specification to a generally recognized standard. Having the SIX standards endorsed by a generally recognized standardization organization and key industry players is likely to aid adoption. A process can be designed to improve the quality of the specification by eliciting feedback from the global expert community and at the same time create support for it in the industry.

1.6 Scientific and social relevance

This research aims to provide a infrastructure designers with a standardized building block to quickly construct multi-party networked service solutions.

Such a standardized solution to a common problem will reduce development costs and time and allow designers to direct resources to innovations in the application domain rather than reinventing solutions to infrastructure, thus potentially leading to more innovative services with higher quality and shorter time-to-market.

Scientifically this research results in an assessment of the suitability of certain technology for with the networked services domain and an approach for performing the assessment.

1.7 Document outline

The rest of the document is structured akin to the following outline. The next chapter within this part, chapter 2, provides the reader with economical and technological background information which will provide a better understanding of the context of the problem and the scientific and social relevance of the research.

The second part of this report documents the further analysis of requirements for this solution with the results of a refinement step in chapter 3 and the construction of a generic solution model in chapter 4.

Part 3 of this report concerns the assessment of existing technologies and discusses the assessment process conducted as part of this research. Chapter 5 introduces the iterative assessment approach and addresses the results of the first two iterations while chapters 6 and 7 each describe the results of a part of the third iteration; the detailed assessment of UDDI (6) and ebXML RS/RIM (2) respectively.

The discussion of the actual specification of the solution is provided in chapter 8 in the fourth part of this document. Readers interested in the main deliverable are referred to this section and Appendix D: SIX:2503 Directory protocol standard (main deliverable) that contain the actual specification. The final part of this document concludes the discussion with a set of conclusions in chapter 9 and a set of recommendation in chapter 10.

2 Background

This section provides the reader with background information on the context and relevance of the research. The first section provides background on the context in which networked services are commonly found: the context of two-sided markets. This section explains the increasing emergence of networked services with economic theory and highlights the relevance of this research from this point of view. The second section approaches the problem from a technological point of view; introducing the evolution of the internet from a communication infrastructure to a transaction infrastructure. It shows the need for standardization of a transaction supporting infrastructure and current lack thereof.

2.1 Economic background

This section provides background on the context of this research from an economical perspective. It highlights the need for networked services from economic point of view. This need for such services in conjunction with the lack of standardization of technology determines the relevance of this research.

Two-sided networks

In so-called two-sided economic networks there are two distinct types of users that require interaction with each other. (Figure 1: Two-sided economic network) These interactions generally take the form of a transaction in which a user of one type initiates the transaction while a user of the other type executes and produces a result.

An example of such a transaction is the online payment of a purchase in an online store. In this example the merchant acts as a user of the first type that initiates the transaction, in this case requests the payment, while the customer is a user of the second type that produces a result, in this case: fulfills the payment.

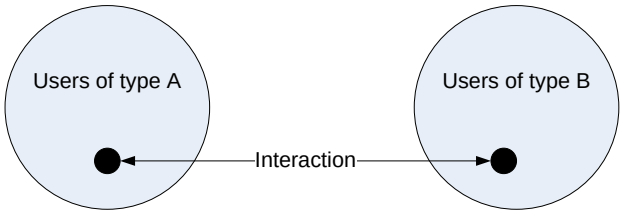

Figure 1: Two-sided economic network

In two-sided economic networks, two kinds of network effects occur: same-side and cross-side effects (7) When users of one type exhibit preference regarding the size of the group of users of the other type this is called the cross-side network effect. The preference regarding the number of users can be both positive and negative. For example in the case of online retail the merchant (user of type A) benefits from a larger number of customers (user of type B) while the customer benefits from a larger number of retailers in terms of price competition and larger supply.

On the other hand the retailer has a preference towards a smaller number of competitors, while the customer might feel positive about an increased number of fellow customers,

increased demand may cause price decrease, or negative, increased demand of a scarce resource may price increase. Users of one type exhibiting preference regarding the size of their peer group is called the same-side effect. (Figure 2: Same-side and cross-side network effects) (7)

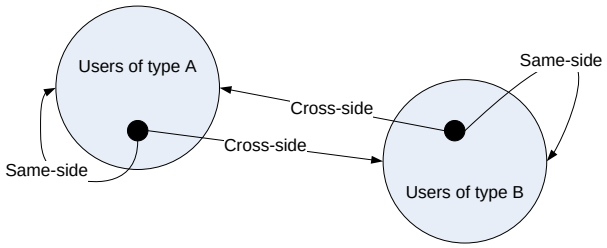

Figure 2: Same-side and cross-side network effects

While online payments provide a compelling example two-sided networks are also found in many other economic areas. In the software industry, for example: end-users and application developers form a two-sided network and in health care: patients and doctors.

The Three-party model

To facilitate the transactions in the two-sided network service providers emerge that position themselves between the users of one type and the users of the other type. That situation conforms to the three-party model: the interaction between users of type A and users of type B is facilitated by a centralized service provider. (8)

In the example of the online payments, the larger online retailers like Amazon.com and Bol.com are big enough to develop and maintain their own online payment infrastructure; however many smaller businesses rely on the services offered by payment service providers. The payment service provider positions itself between the customer and the merchant facilitating a transaction between both parties and offering services to both side of the economic network. PayPal, for example, is such a centralized service provider. (9)

Economic optimization and the network effects of the three-party model network lead after an initial period of competition between service providers (and networks) to the establishment of a single network that connects all users. (10) As a consequence all power transfers to a single dominant service provider. There are also many other cases where a network was designed according to the three-party model with a single centralized service provider.

This effect is illustrated by this example in software development. An operating system can be considered the central platform in a three party network, connecting software developers and software users. Let assume the market for operating systems is relatively new and multiple competitors exist. Both software developers and software users need to select the platform to develop their software for and run their systems on respectively. Obviously a software vendor would like to sell its software to as many users as possible,

while a user would like to have as much choice as possible in selecting its software. The user exhibits a positive preference regarding the number of vendors on the platform while the vendor exhibits a positive preference regarding the number of users of the platform. Initially the different platforms are limited in size and compete on available software, speed, functionality and other qualitative aspects. However once a platform reaches a certain number of users of both types its size (number of users) becomes an property platforms compete on.

In online international payments PayPal has become the dominant service provider, but a more compelling example from software technology is Microsoft Windows. Microsoft has positioned its operating system between end-users and application developers; optimization of the market has resulted in an extremely dominant position for this platform.

The economic and network effects of a three-party model network interfere with liberal economic principles, such as free competition. Therefore the 3-party model faces increasing resistance from both governments and businesses. (9) (11) (12)

Four-party[1] networks

Because of the increased market and regulatory resistance against three-party or centralized solutions, an increasing number of networks with a four-party model is created. In the four-party model the role of the centralized service provider is split in separate roles for service providers that provide services to either one of the user types in the network. Due to this split either of these roles can now be played by any number of service providers.

For this analysis we are looking at multi-party networks, with at least four parties (users and service providers) that are involved in performing a single interaction. Contrary to three-party networks (FIGURE 3a) within this type of network there is no centralized entity, or platform, that connects users of different types, instead all service providers provide services to users of a single user type and service providers are bilaterally connected to each other. (FIGURE 3b)

[1] May also be more than four parties

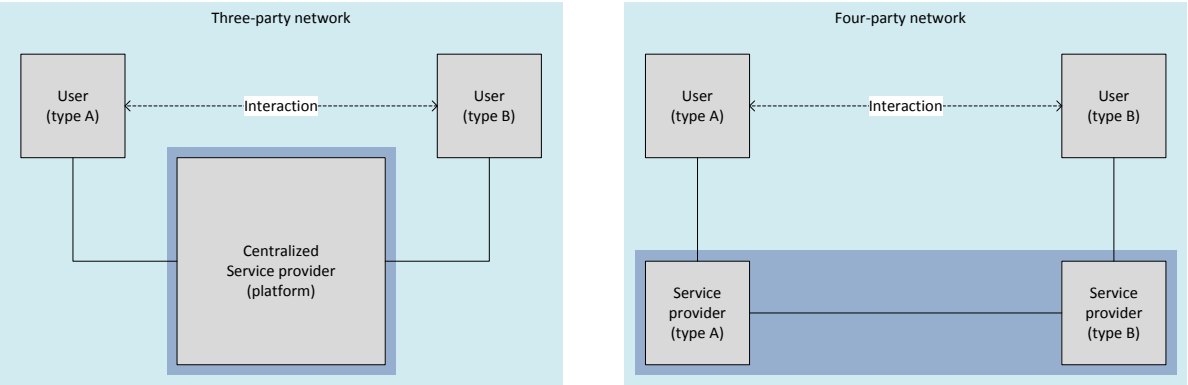

Figure 3: Three-Party Network (A) And Four-Party Network (B)

Within this network the parties exchange data. We limit our scope to situation in which this data exchange represents (part of) a business transaction that represents real-world value to the parties involved; for example the transfer of money or the digital signing of a contract. The value that these transactions represent demands high levels of security, reliability and trust in the authenticity and integrity of the information exchanged.

The size of the multi-party networks can range between a few participants up to several hundreds of participants. Participants can be geographically located anywhere in the world.

In the online payment industry PayPal is an example of a centralized service provider in a three-party model network. PayPal has positioned itself in such a way that it dominates the international online payments market. Another approach was taken with iDEAL (4), an online payment standard developed and employed by the major Dutch banks. iDEAL is designed in such a way that no centralized service provider is needed. Instead the two roles generally played by the centralized service provider in a three-party model are played by different actors. The role that deals with the merchants, the acquirer role (which acquires payment for the merchant) and another role that deals with the customers, the issuer role (which issued the customers payment method).

In the four-party model networks the interaction between all parties must be governed by a set of rules and regulations that safeguard free competition by creating a 'level playing field'. The level playing field is a fairness principle dictating that each 'player', party in the network, plays by the same rules, i.e. there are no regulations that affect the ability of players to compete fairly. This situation is unlike the three-party model in which a limited number (generally just a single one) of centralized service providers becomes powerful enough to dictate the 'rules'. These rules and regulation also guide the interaction between the 'players'.

Within these networks more than one party can offer services and all parties compete with their own distinct proposition. An important characteristic of this type of network is

the equal distribution of power across the network and the fact that all users profit from network growth while allowing free competition among the parties. (7)

Schemes

The network can be governed by a scheme. A scheme is a set of agreements, rules and regulations, and a selection of standards that create a level playing field for all participants. The scheme creates a cooperative domain which ensures that all participants can cooperate successfully and it creates a competitive domain that ensures all participants can compete fairly at the same time.

The scheme can be said to consist of different layers: a typical scheme consists of the following layer (1).

- In the 'participants and proposition layer' all participants have the freedom to create their own propositions for the services they provide. They can decide on their own business model, revenue models, pricing strategy, target market, etc. In the example of iDEAL one of the participating banks targets its acquiring services at the top 50 large retailers while another participant aims to provide an affordable service to small and medium businesses. . Aspects in this layer are deliberately not regulated by the scheme to allow free completion between participants.
- The 'business/governance layer' of a scheme deals with issues such as: branding, licensing, entrance rules and certification. All these issues are regulated by the scheme.
- The 'application layer' deals with issues typical for the application domain of the scheme. These issues can include: product features and message standards. All these issues are regulated by the scheme.
- The 'infrastructure layer' deals with the low level technical infrastructural issues such as: protocols, security and connectivity. All these issues are regulated by the scheme.

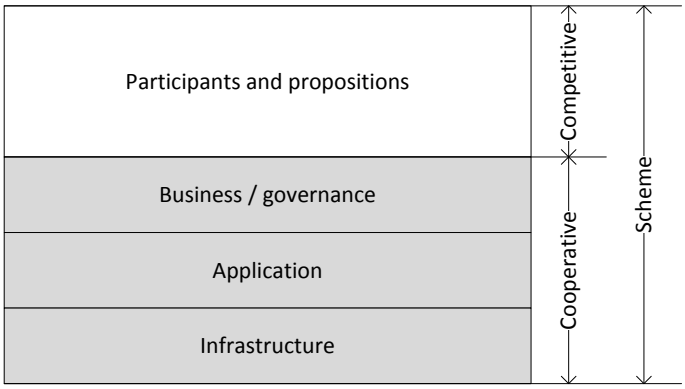

Figure 4: Layers of a scheme

The 'participants and proposition layer' can be considered the competitive domain, in which the participants compete with each other, while the other three layers together make up the cooperative domain, in which the participants cooperate.

A scheme is generally governed by an entity, the scheme organization. The scheme organization maintains the agreements and specifications, including infrastructural service and interface specifications. It also regulates the joining and leaving of participants on business/governance level. This organization is not a market party but an independent) (not-for-profit) organization.

2.2 Technological background

This section provides background on the technological relevance of this research. It describes the transition of the internet from a communication infrastructure towards a transaction infrastructure and the evolvement of internet technology along with it.

In short the current ubiquity of internet connectivity enables and promotes this network as transaction channel for all sorts of e-commerce transactions, including: payments, invoicing, procurement and others.

The traditional Internet technology stack (13) consisting of: TCP/IP (14) (15), generally in combination with a popular application layer protocol such as HTTP (16), SMTP (17) and FTP (18). Optionally in combination with transport layer security protocols such as SSL (19) and TLS (20) does not meet all requirements for these infrastructures for online networked services. In addition to the traditional internet technology stack, many technologies, protocols and standards have been drafted to meet (some of) these requirements.

These additional technologies include what is called the core web services technology stack: SOAP (21) and WSDL (22)and UDDI (6) that enable structured message exchange, interface specification and service discovery. On top of these web service technologies many technologies have been developed to solve certain shortcomings the still exist with this infrastructure. Examples include: WS-Security (23) these specifies security related protocol extensions to SOAP, WS-ReliableMessaging (24) that specifies protocol extensions to SOAP that improve or guarantee the reliability of the SOAP message exchange and WS-Addressing (25) that specifies SOAP headers that contain addressing information.

2.2.1 Lack of standardization

The plethora of existing specifications still leaves many options open for infrastructure developers that need to understand and consider each technology. While this provides freedom to infrastructure developers to select technologies that best suit their needs it also introduces interoperability issues. WS-Interoperability (26) is an initiative to alleviate this problem by grouping certain sets of specifications into profiles. For example the WS-I Basic profile or the WS-I Secure basic profile (27). While WS-I is a step forward, it still leaves room for a lot of alternatives and still requires many choices to be made.

Because of the complexity of the requirements and the many choices infrastructure development for networked services is expensive. Assessing and selecting the appropriate technologies, specification is a time-consuming and thus expensive task in

terms of lead-time and man hours in projects. Yet for each initiative it is redone. Just in the Netherlands alone the initiatives that Innopay has been involved in, including iDEAL (4), Standaard Digitale Nota, all have set out to develop their protocols from scratch. The problem is recognized at international level by the major financial institutions in the European Payments Counsil that set out to develop the e-operating model (28) Also the United Nations Centre for Trade Facilitation and Electronic Business has started with the development of the ebXML specification (2).

The current practice of 'reinventing the wheel' is undesirable for a number of reasons and standardization is needed. The most obvious reason to support standardization is the saving of costs and time by having a ready-to-use building block available to infrastructure developers that is proven to be well suited for the required infrastructure.

Secondly, standardization will improve the innovation power across the board because less attention and resources need to be devoted to infrastructure development and can be focused on the areas where innovation happens; i.e. on application and business level. With any infrastructure that has been highly standardized, for example the electricity grid, innovation takes place in the application of the technology, rather than in the infrastructure.

2.3 SIX standard

Many electronic (online) services use the four-party model and a scheme approach to create and regulate the transaction network. All participants in this network need to exchange (transactional) data. Electronic four-party schemes require an infrastructure that supports a scalable network of participants and secure information exchange between participants.

Currently, each e-scheme designs its own infrastructure which is time and effort consuming but adds no direct value to the end-user. The availability of a generic messaging layer standard that restricts the e-scheme developer's freedom of design would significantly reduce cost and risk of developing an e-scheme and would speed up e-scheme development and implementation drastically.

To fulfill this need Innopay has started the development of a standard to cover infrastructural aspects of e-schemes: the Secure Information eXchange standard (SIX standard).

2.3.1 SIX standard

To provide a standardized building block for the infrastructure of such networks Innopay is developing the Secure Information eXchange standard (SIX). The scope of the SIX standard is defined as follows: SIX strives to create a generic, reusable set of standards to facilitate: **multi-party electronic schemes** that offer **real-time (online) services**, for which **transaction related data** needs to be **exchanged** in a **secure** manner. (1)

The set of standards combined can be used to create the infrastructure layer appropriate for the application of the scheme. This requires the messaging layer to be generic and application agnostic within the scope of the e-schemes.

The SIX standard is comprised of 5 (sub-)standards:

SIX:0208 Secure SOAP Interface standard
The secure SOAP interface standard defines web services secure messaging between participants in the network. It is entirely based on generally accepted internet messaging and security standards, including: HTTP, SOAP, XML-Signature, XML-Encryption, WS-Security, WS-Policy, WS-SecurityPolicy, WS-Addressing, SHA-256, asymmetric RSA encryption, X.509 and WSDL. This substandard is fully developed.

SIX:1712 Secure ASN.1 Interface standard
This substandard defines secure messaging based on ASN.1 (Abstract Syntax Notation) (29) data structures , commonly used in point-of-sale and ATM terminals.

SIX:2909: Real-time Online Protocol
The SIX standards are developed with two-sided real-time multi-party networks in mind. This substandard defines how SIX:0208 Secure SOAP interface standard and SIX:2503 Directory protocol standard may be combined to create an addressable network with message flows across multiple participants. Such an addressable network and message flow definition form the basis for the infrastructure layer of an e-scheme.

SIX:3007 Real-time PoS Protocol
The SIX standards are developed with two-sided real-time multi-party networks in mind. This substandard defines how SIX:0208 Secure SOAP interface standard, SIX:1712 Secure ASN.1 Interface standard and SIX:2503 Directory protocol standard may be combined to create an addressable network with message flows across multiple participants. Such an addressable network and message flow definition form the basis for the infrastructure layer of an e-scheme.

SIX:2503 Directory protocol
Since in a multi-party network participants may interact with all other participating parties, they must be able to connect to and authenticate any participating party. It is essential that communication, availability and identity information of all the participating parties is available to all the participating parties. The Directory protocol defines how to create and maintain directories of this information. This substandard has not been created yet and will be the subject of my research, with drafting of this standard as its main goal.

Furthermore Innopay has set out with the following principles for the SIX standards:

Reusability
Inventing, settings up and maintaining a messaging standard is time and effort consuming: the use of readily available components reduces costs and development time.

Scalability
Most e-schemes have the ambition to grow their network. The standard needs to be able to cope with large numbers of participants.

Security
Messages sent within e-schemes concern transactions that may have legal implications, stringent security measures are required.

PART 2 ANALYSING REQUIREMENTS

This part discusses the first step in the research process: the analysis and refinement of the requirements and the translation of those requirements into generic solution models and mechanisms.

3 Requirements refinement

This section discusses the first step of this research: a refinement of the results of the requirements analysis. The results of the requirements analysis are taken from (3) which is available in Appendix A: Requirements overview. Section 3.1 discusses the requirements analysis result of the analysis performed as part of the preliminary research for this thesis project. The second section describes the subsequent refinement process.

3.1 Requirements analysis

In (3) the results of a requirements analysis are discussed. The documented research consisted of a requirements elicitation and analysis as well as a technology survey. It yielded a set of requirements and a set of technologies that potentially provide (partial) solutions to the information distribution project. Both were conducted to serve as input for this research project.

Ax described in (3) this set of requirements is the results of an analysis process and based on the input of experienced expert and inspection of existing solutions.

The goal of the requirements analysis was to obtain a set of requirements that could guide the technology survey in terms of direction, types of technology, to explore and to provide a means to decide which technologies devote attention to and which to ignore. It was determined that the results could be refined if necessary for further research.

The requirements elicitation was based on expert opinion, the input from an expert group, consisting of 5 experienced senior consultants involved in the development of more than one networked service, and case study and analysis of recent networked service developments. Input has been elicited during interviews and 2 brainstorm sessions. From these inputs a context model has been constructed of which use cases have been derived. With this information the analysis yielded a set of requirements related to the protocol and the specification.

The requirements analysis yielded a set of 33 (high-level) requirements. It was concluded in (3) that the level of detail was suitable for conducting the technology survey, however that further refinement was needed for structured assessment of technologies. Areas for additional research and specification have also been determined, they include:

- Methods for secure certificate distribution
- Methods for real-time notification

These areas will receive more attention during the refinement.

3.2 Requirements refinement

To be able to conduct the technology assessment later on in this research and to aid the drafting of the specification the requirements uncovered in the research describe above need to be refined.

The goal of the refinement step is to determine a set of requirements that, when met by the specification, ensure that technology according to that specification is a solution to the data distribution problem this research project is attempting to find a solution for. Practically this means that the requirements must be able to be used as criteria in the technology assessment and selection to be conducted as part of this project and assist the drafting of the specification based on the selected technology.

The requirements refinement adds additional requirements to the existing set. The additional requirements will never contradict an existing statement. Some additional statements increase the level of detail of an existing statement. For example, the statement: 'REQ3: The directory service specified by the standard MUST retain authentication information,' will be made more specific by including statements that specify what 'authentication information' consists of.

Other additional statements will decrease the abstraction level of the statements. The level of abstraction can be high, i.e. very abstract, generic, not related to implementation specific details, or low, relating to issues on an implementation specific level. The statement in requirement REQ3 above could be refined with a statement specifying the format in which authentication information is to be specified.

The input for the additional statements is taken from the same sources as the original requirements analysis: expert opinion and case study. Additionally the existing parts of the SIX standards specification are consulted for technical details on the communication protocol used between service providers.

The refinement of requirements focuses on the following areas:

- Information model requirements
- Access control requirements
- Functional requirements, specifically related to the real-time data distribution
- Interface requirements

The results of the refinement process are documented in appendix B. The process has yielded the specification of 19 requirements in addition to the existing high-level requirements that more specifically or less abstractly limit design freedom.

These requirements are in fact requirements on the set of requirements that will make up the directory protocol specification. To be read as: the specification must, should or could require or recommend a certain aspect. While these requirements prescribes properties on the eventual solution the end result of this research will not be an actual solution, but again a specification of (requirements on) such a solution.

4 Generic models and mechanisms

To aid the accurate assessment of relevant technologies in the next section generic models of certain aspects of the solution are constructed. The generic models can later be mapped onto a selected technology to assess the fit in more detail. The mapping of a generic model onto a selected technology will enable assessment of the amount of effort necessary to let the selected technology meet the requirements.

4.1 Approach

The generic models are constructed based on the requirements and refined requirements that can be found in appendix A and B.

The requirements prescribe certain characteristics of the solution; mostly functional aspects. Two sets of requirements are translated into generic models: the information model requirements and the access control model requirements.

From the information model requirements a generic information model that would satisfy these requirements is constructed. The information model requirements can now be replaced by the generic model plus the new requirement stating the technology must be able to support the generic information model. Hence, a mapping between the generic information model and the native information model of the technology must exist.

From the requirements relating to the access control model a generic access control mechanism or model is constructed. An access control model consists of a set of rules that link together Subjects (e.g. users), Resources (e.g. documents) and Operations (e.g. read, write) (30). In the detailed assessment for each of these rules it can be determined whether the technology can enforce it.

4.2 Generic data model

In this section a generic data model is constructed.

From the requirements specification it becomes clear that there are two sets of data: participant related data and general scheme related data. The participant-related data consists of information related to the participant's business entity or the services it delivers. The requirements specification lists 4 categories of information:

- Identification information
- Addressing information
- Authentication information
- Availability information

The identification information consists of all information other participants need in order to identify a participant. From the low-level requirement specifications we take the set of information as consisting of:

- a participant identifier, specific to the scheme,
- a name for the participant,

- one or more identifiers for the role(s) the participant performs in the scheme,
- contact information for the participant consisting of a contact name, phone number and e-mail address.

Thus, we define a basic Participant entity with the following properties.

Property	**Format**
Participant identifier	Any non-empty string (format to be specified by scheme)
Participant name	Any non-empty string
Contact person name	Any non-empty string
Contact e-mail address	Any valid e-mail address
Contact phone number	Any valid international phone number

Table 1: Contact information properties

Each participant performs one of more roles. For each role a specific service is provided. For each service a specification should be provided. The service related information consists of the addressing, authentication and availability information.

- Addressing information is provided in the form of service implementation specification documents adhering to the WSDL specification (22) and WS-SecurityPolicy (31) documents.
- Authentication information is a collection of certificates. The XML Digital signature specification (32) provides a XML data format for such certificates.
- Availability information consists of both the current state and planned unavailability intervals. There is no specification available and so the format is specified below.

Figure 5 provides an overview of the generic data model as described above. UML class diagram notation is used. Data types, association identifiers are omitted for readability.

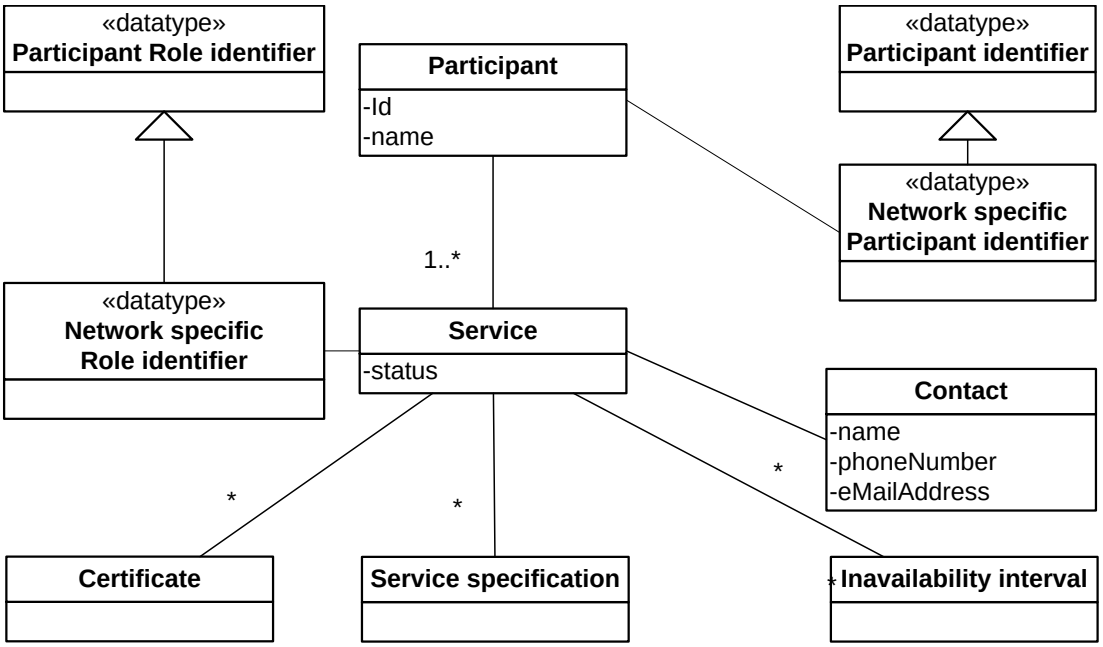

Figure 5: Generic data model

4.2.1 Availability information

The availability information mentioned in the requirements analysis consists of two types of information. A real-time status and planned (un)availability intervals.

- real-time status which can be expressed by a single value indicating current service status.
- unavailability interval which can be specified by an expected start and end time.

4.2.2 Non-participant related data

Next to the data relating to specific participants there are information items that relate to the scheme or network as a whole. Non-participant related data consists of service interface specifications, scheme role definitions.

- Participant roles scheme
- Service interface definitions.

Service interface definitions are specification of the interfaces used in the scheme. The specification are determined by the scheme organization and participants are required by scheme rules to implement interface according to these specifications. The service interface specifications are specified using WSDL documents and WS-SecurityPolicy documents.

The participants in a scheme all perform one or more roles. A proper designation for these roles must be defined. This set of designations must be published by the scheme organization.

4.3 Generic access control model

The access control related requirements in the requirements analysis could be fulfilled with the following generic specification. The generic solution uses a role-based access control approach (33) as well as the concept of information ownership.

With role-based access control all users are associated with zero or more roles. Based on these roles associated with the user and the access rules that use these roles access to a resource by a user is granted or denied.

The concept of information ownership means that every object is associated with a user. This user is considered the owner of the object and depending on the access control policy may have special privileges regarding the object.

An access control policy can be defined by specify by permitting or preventing users (subjects) from executing a specific action (operation) on specific information objects (resources).

We define the following roles for access control purposes:

- Scheme managing
- Participating

Next we specify ownership for the specific data elements:

- Non-participant related data is owned by the scheme organization.
- Participant related data is owned by the participant with the exception of participant type data and the participant identifier, the latter are owned by the scheme organization.

The solution must enforce the following access rules.

1. Users in the participating of scheme managing role can read all resources
2. Owners can modify or delete all resources they own.
3. Users in the scheme organizing role can create Participant objects
4. Users in the scheme organizing role can create non-participant related data
5. Users in the participating role can create resources referencing resources they own.

PART 3 ASSESSING EXISTING TECHNOLOGIES

A first step towards a solution is the assessment existing specifications, technologies and solutions. A technology survey resulted in a set of existing technologies of which the suitability to function as a basis for the directory protocol specification is determined. These technologies are now submitted to a more elaborate assessment.

5 Technology Assessment

This chapter discusses the assessment and selection process and results of existing specifications/technologies.

5.1 Assessment approach

In this section the assessment process is outlined. The goal of this assessment is to find the technology for which it is most likely that a solution based on that technology provides the best solution. The best solution is a solution that meets all requirements with the least specification effort. With the technology that is found with this assessment a proof-of-concept of the solution is constructed.

Because it is infeasible to construct proof-of-concepts for each technology, the proof-of-concept based on the technology found by the assessment will be selected for use in the specification if the constructed the proof-of-concept meets all requirements. Alternatively the outcome of the assessment may be that none of the included technologies is sufficiently likely to provide a good basis.

Iterations

The assessment is conducted in an iterative process. With each iteration the level of depth of the assessment is increased.

The assessment process will go through the following iterations:

- The first iteration in the assessment process deals with meta-aspects of the technologies, such as: standardization, maturity, use and usage. With a set of cut-off criteria a number of technologies will be eliminated from the next iteration of the process.

- The second iteration will deal with more detailed functional aspects of the technologies under consideration. For the assessment the technologies and standards included will have to be looked at in more detail. Criteria used in the phase are for example the degree of support for bulk updates or access control mechanisms. With these criteria the technologies are scored and a ranking is determined rating the solutions from most likely to provide an optimal solution to least likely.

- A third phase of the assessment involves the actual mapping of the requirements on the features of the technology. This is process involves detailed study of the solution and thus is very time-consuming. For this reason it is only conducted for the highest ranking solution. If it determined during this process that the technology will not meet all requirements the decision can be made to abandon it and move on with the next highest ranking technology.

Scoring

The method of scoring of a technology or solution on a criterion depends on the nature of the criterion and could range from strictly quantative to strictly qualitative. The method of assigning score will be determined for each individual criterion.

5.2 First iteration

The first iteration in the assessment process deals with meta-aspects of the technologies. This section discusses this assessment stage. Section 5.2.1 introduces the technologies included in the assessment. Section 5.2.2discusses the criteria use in this assessment phase. The assessment results can be found in section 5.2.3.

5.2.1 Technologies

The following technologies are included in this iteration. These technologies were found in the technology survey in (3)

ebXML Registry and repository service

The ebXML Registry Specification (2) includes both a mechanism for publishing service specification meta-data as well as storing and retrieving the actual documents describing the services. The ebXML RS specification is accompanied by the ebXML Registry information model specification (34) that defines information model used by compliant registries. Besides these registry related specification the ebXML suite, designed to enable a global electronic market, includes a messaging service specification, a collaboration protocol specification and a business process specification. ebXML is developed and maintained by OASIS and UN/CEFACT. For this assessment version 3.0 of the ebXML specification is considered.

Lightweight Directory Access Protocol (LDAP) and Directory Service Markup Language (DSML)

The LDAP (35) specification defines a general-purpose directory system and service that can store and retrieve any type of data in/from its tree-based data structure. LDAP is not specifically designed for service specification registry but can store any type of data. The LDAP specification is developed and maintained by the IETF. We consider version 3 of the LDAP specification. Most modern implementations of LDAP include a DSML (36) interface that allows XML (37) formatted messages for interacting with the LDAP service. These standards are considered in conjunction with each other.

MuleSoft MuleGalaxy

MuleGalaxy (38) is a product that allows the publication and manipulation of service specifications and related meta-data. It is developed and maintained by MuleSoft, a software vendor of an open-source Enterprise Service Bus solution. MuleGalaxy is currently at version 1.5.

Universal Description Discovery and Integration (UDDI)

The UDDI registry specification (6) was designed to provide a means for storing web service specification meta-data. Together with the SOAP and WSDL specification it forms

the core of the original web services specifications stack. The specification is developed and maintained by OASIS. Version 4 is currently under development but version 3.0 of the specification is considered in this assessment.

WSO2 Governance registry
This part of the WSO2 (39) open-source SOA middleware solution developed and maintained by WSO2 can maintain a library of service specifications and related documents. It exposes these resources through a REST-style (40) interface.

XML Key Management Service specification (XKMS):
The XKMS (41) specification defines a protocol for interacting with a key store. The message format of the protocol is XML-based. The specification is developed and maintained by the W3C. We look at XKMS version 2.0.

Tailor-made solution
An alternative to using an existing technology is designing something entirely new, specifically tailored towards the requirements. Such a solution would by definition meet all functional requirements and is not limited by the existence of implementations. However it also does not benefit from reuse of existing technology or have the support of a mayor standardization body.

5.2.2 Criteria

The following criteria are used in this assessment iteration.

Standardisation aspects
For assessing the standardization the following factors are taken into account:

Is the technology a standard?
We consider only specifications that are published as a standard or recommendation by a generally recognized standardization body and are implemented by at least more than vendor or (open-source) project. The interface of a product from a single vendor cannot be considered a standard. A technology that is standardized by a recognized standardization institute is favoured over technologies that are not. Basing the standard to be created on existing standards helps adoption of the new standard.

Involved parties: who owns, maintains, supports and implements the technology/standard
In relation with the aspect above we look at the parties involved with the technology in various roles. The standardization institute is involved and the parties maintaining and contributing to the development of the standard are relevant. A technology that has the support of 'big' names is beneficiary to the adoption of the new standard.

Maturity: how mature is the technology/standard?
Another relevant aspect is the level of maturity of the technology. Relevant factors include the age of the technology, the fact that it is still under active development and the version it is in. A more mature standard is considered better.

Use and usage aspects

Use and usage, both qualitative (how) and quantitative (how many), are relevant aspects for this assessment. The quantitative aspect of use is relevant for the ease of adoption. Technologies that are in common use are more easily adopted that relatively little used technologies. However, when considering the use of a technology the common usage (how it is used) should be taken into account. Using a technology for a purpose for which it is commonly used (because it was designed for that task) will probably provide a better fit between the technology and the requirements and help adoption.

It can be difficult to determine and quantify the actual use of a technology therefore the use can be indicated by other metrics, such as: the number of implementations, the number of downloads of implementations and the number of relevant search engine results.

Functionality

We also take a high-level look at the functionality offered by the technology. Later phases will contain a more detailed look at functionality but in this phase we consider the following two aspects:

Storing of meta-data and actual documents

There are two types of relevant data to be stored in the directory: documents describing the various aspects of the services and meta-data that enables users of the directory to retrieve the relevant documents.

Support for real-time notification

A key requirement for the directory system is the ability for users to be actively notified of relevant updates to the data in the system.

5.2.3 Conclusions

Table 2 provides an overview of the scores of the different technologies on the various criteria. An elaboration of the assessment results can be found in appendix C.

	ebXML	LDAP	Mule	UDDI	WSO2	XKMS	Tailor-made
Standardization	+	++	--	++	--	+	--
Use/usage	-	++	-	++	-	o	-
Functionality	++	+	+	+	+	o	++

Table 2: Assessment results first iteration

Selected alternatives

The most promising alternatives are UDDI, ebXML and LDAP. Also strong score on functionality of the tailor-made solution can be noticed. These technologies are selected for further assessment in the next iteration.

Discarded alternatives

The aim of this project is to create an open standard based on existing open standards. Both WSO2 Governance registry and MuleGalaxy are technological solutions to the problem but the interfaces of these products are not based on a standard that are

maintained and supported by one of the major standardization organisations or backed by major software vendors and hence it is undesirable to base the SIX:2503 directory protocol standard on any of these products.

The coverage of the functional requirements of the XKMS interface is very limited. Only the authentication related information subset is covered. A solution involving XMKS therefore has to include other technologies. We discard XKMS as a stand-alone option but may consider it in a solution based on other technologies.

5.3 Second iteration

This section discusses the second assessment iteration. This stage focuses on the functional requirements.

5.3.1 Technologies

In this second iteration the following technologies are assessed.

- A UDDI registry provides a means for storing service specification meta-data. However the UDDI specification does not include repository functionality so for storage of the actual specification documents additional technology needs to be included. In the most simple form this is a standard web server that allows resource manipulation through default HTTP operations. Version 3.0 of the UDDI specification is considered.
- ebXML registry and repository service is included in the assessment without additional specification.
- LDAP is considered in combination with the DSML interface specification.
- A tailor-made alternative to using an existing technology is creating something entirely new, specifically tailored towards the requirements.

5.3.2 Criteria

In this assessment phase the following list of criteria will be used. The list is a specialisation of the functionality criterion used in the first phase of the assessment.

- Data model compatibility: the data model compatibility consists of two factors: the ability of the solution to store all needed information: identification information, service specification documents, public keys, availability information and on the other hand the amount of effort it would take to specify the data model or adapt the existing data model of the solution.
- The functionality provided by the solutions is compared with the required functionality with respect to the following aspects.
    - Single participant query and other basic operations, such as update and delete.
    - Synchronization: does the solution provide a way to synchronize local (cached) copies of the data set with its data set.
    - Audit trails and lifecycle management: does the solution record changes to the directory contents and does it have support for versioning of its contents

    - Notification mechanism: is there a subscription-based interface that lets participants subscribe to relevant notification or another mechanism for distributing information in real-time
- Access control mechanism: is there a configurable access control mechanism that support the access control rules as specified by the requirements.
- Interface: the solution should support an interface that complies to the SIX:0208 Secure SOAP interface specification. But if it does not an XML-based protocol over HTTP connections is preferred (REQ28).

Weights

Not all criteria are of equal importance. For this reason we assign different weights for each criterion. The assigned weights are determined during an expert group session.

5.3.3 Conclusions

This section contains the conclusions of the assessment. An elaboration of the results can be found in appendix D.

The following functional criteria result in the revised score on functionality.

	Weight	**ebXML**	**LDAP**	**UDDI**	**Tailor-made**
Simple CRUD	10	1	0	2	2
Synchronization	5	1	1	2	2
Audit trails	3	2	-1	0	2
Notifications	5	1	-1	1	2
Scores		**1,1**	**-0,1**	**1,52**	**2**

Table 3: Assessment results second iteration

If we include this revised scores on functionality in the assessment with the other criteria, we get the results as in .

	Weight	**ebXML**	**LDAP**	**UDDI**	**Tailor-made**
Standardization	10	1	2	2	-2
Use/usage	10	-1	2	2	-2
Data-model	8	1	-1	1	2
Functionality	*10*	*1,1*	*-0,1*	*1,52*	*2*
Access control mechanism	8	2	0	1	2
Interface	5	1	-2	2	2
Score		**0.79**	**0.41**	**1.59**	**0.43**

Table 4: Assessment results second iteration

Ranking

The following ranks are attributed to solutions (higher is more likely to be a good fit)

1. UDDI
2. ebXML RS
3. tailor-made
4. LDAP

The most promising solution is UDDI registry in combination with a simple centralized repository. UDDI was designed as a service specification distribution solution. Although it is only a registry and additional technologies need to be selected to complete the solution this third version offers a big part of the desired functionality, has a very compatible data model. It also is a proper standard with wide industry support. It support different access control policies and its interface is based on SOAP.

ebXML registry and repository offers a more generic solution. Not specifically designed to distribute services specification but aimed at distributing any type of document it includes the repository functionality the UDDI lacks, in that respect ebXML has the advantage of a more integrated solution, because of the build in repository. The major shortcoming of ebXML is the lack of maturity and limited use.

A tailor-made solution would satisfy all functional criteria, but the issues regarding standardization and specification effort still hold.

LDAP, while having a strong presence, has a less compatible data model and interfaces. The DSML interface does provide SOAP and XML based front-end but should be considered a workaround for connecting legacy systems to modern SOA infrastructures. Another disadvantage of LDAP is the fact that the specification does not cover all aspects and lots of implementation-specific (non-standardized) functionality exists.

6 Proof of concept: Universal Description Discovery Integration

In this chapter the use of UDDI as a basis for the directory protocol specification is investigated in more depth. UDDI received the highest score in the second assessment iteration. It scored high on standardization and use/usage. Functionally it appears to provide adequate basic functionality and to have a compatible interface. Detailed examination of the specification in conjunction with the refined requirements and generic models resulted in the following observations.

6.1 Data model mapping

This section describes the mapping of the generic data model on the UDDI data model. The mapping is generally of a syntactic nature, any semantic resemblance between concepts is purely coincidental. The relevant UDDI constructs are documented in **Error! Reference source not found.**.

6.1.1 Identification information

The participant is the primary container in the requirements data model and as such is mapped to the UDDI Business Entity that forms the top element of the UDDI data hierarchy.

Participant Identifier

UDDI provides a mechanism for assigning multiple identifiers to a business entity. This mechanism consists of an 'Identifier bag' that can contain multiple references to identifiers and the identifier system they belong to. For the solution each scheme should define its own scheme-specific identifier system that is however denoted as a participant identifier system. The identifier system can be specified with a UDDI tModel, a general construct in UDDI that can respresent among other thing an identifier system. The standard must require the scheme developer to specify a tModel for the scheme specific identifier system.

UDDI provides a mechanism for categorizing entities, including tModels, similar to the identifier bag. Category bags are used to contain any number of references to categorizations from different categorization systems. Similar to identifier systems these categorization systems are specified by tModels. The scheme specific identifier system tModel's category bag must contain a reference to a tModel categorizing it as a participant identifier. The standard should prescribe this tModel definition and require the scheme specific tModel's category bag to contain a reference to the aforementioned tModel.

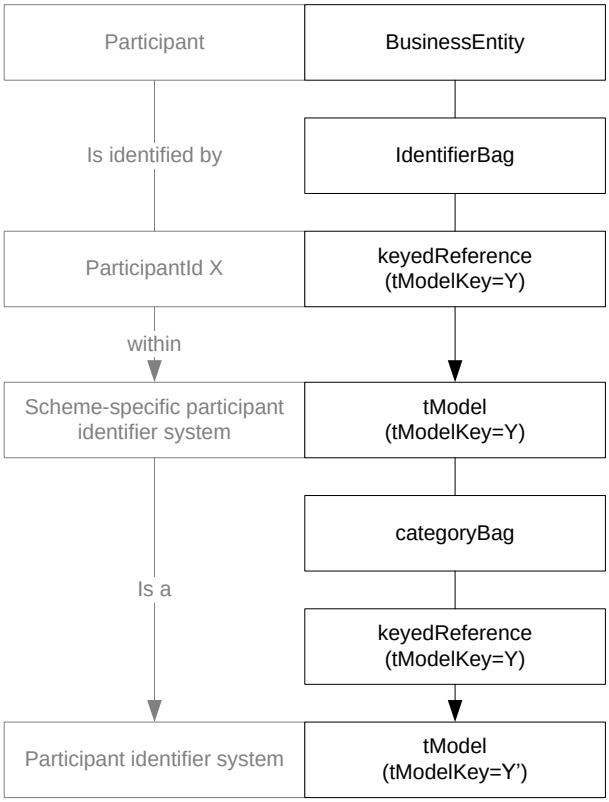

Figure 6: UDDI data model mapping (participant identification) Generic data model concepts on the left (grey), mapped to UDDI concepts (black)

Participant contact information

The participant contact information can be stored with the relevant UDDI concepts.

Participant role

The UDDI data model allows a Business entity to include a number of business service specifications that contain a business description of the services offered by the entity. The same categorization mechanism is provided for service as is present for entities and this can be used to represent the participant type. For each role a business service can be defined and that service can be categorized using a scheme specific categorization system that defines all roles in context of the scheme. The categorization system is represented using a tModel that is referenced from the category bag of the service specification. Similar to the participant identifier system tModel this tModel should be categorized as denoting a scheme role.

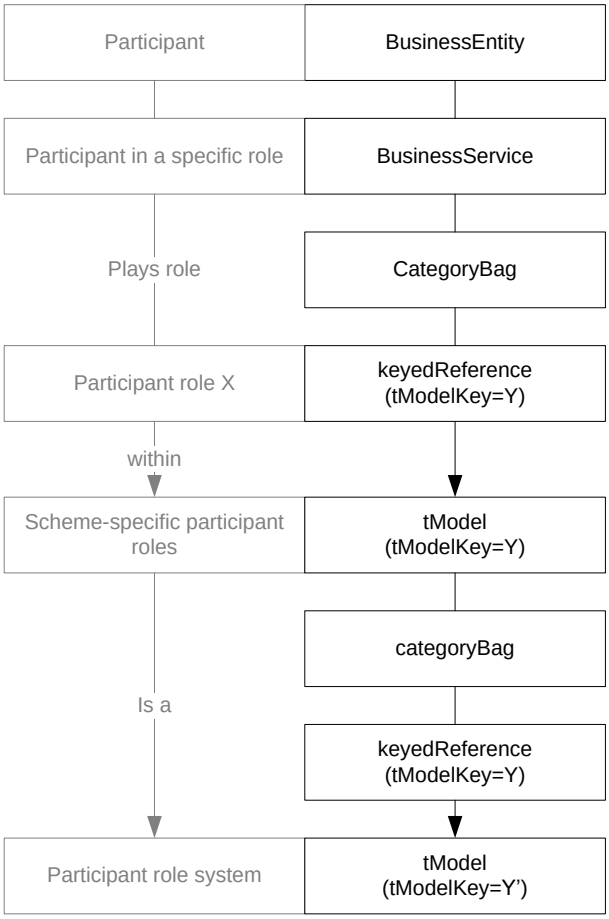

Figure 7: UDDI data model mapping (participant role) Generic data model concepts on the left (grey), mapped to UDDI concepts (black)

6.1.2 Addressing information

This section discusses the mapping of the addressing information related data model.

Services

Associated with each scheme role is a service provided by the participant. The technical aspects of this service can be captured under the UDDI binding template. The different technical aspects of the service, such as authentication and addressing information can be documented using tModelInstanceDetails.

WSDL

The service addressing information is captured in a WSDL document. WSDL is an XML-based format for describing service interface and service implementation specification. Each service provided by the participant has an implementation specification that indicates how the service can be accessed and methods can be invoked. The implementation specification refers to a service interface specification that is registered in participant-independently by the scheme organization. This ensures that all participants conform to the specified service interface.

The WSDL document can be captured inside the instanceParams-element of the instanceDetails-element of the tModelInstanceDetails-element in the UDDI binding template. The tModelInstanceDetails-element has a tModelKey-attribute that should reference the tModel that indicates this element contains addressing information. The standard should specify the tModel used for this purpose.

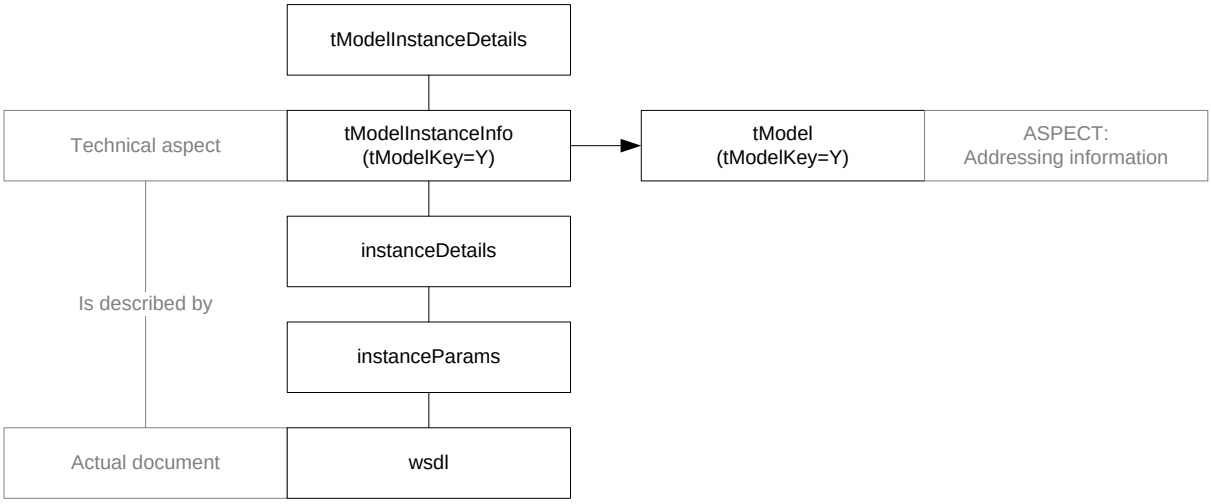

Figure 8: UDDI data model mapping (addressing information) Generic data model concepts on the left (grey), mapped to UDDI concepts (black)

6.1.3 Authentication information

This section discusses the mapping of the authentication information related data model.

Certificates

The requirements specify that multiple certificates must be associated with a participant in a specific role. The selected data format for the certificates is taken from the XML Digital signature specification (32) The KeyInfo container from this specification can be embedded in a tModelINstanceInfo element. This element must contain a reference to a tModel indicating this is a reference to a certificate.

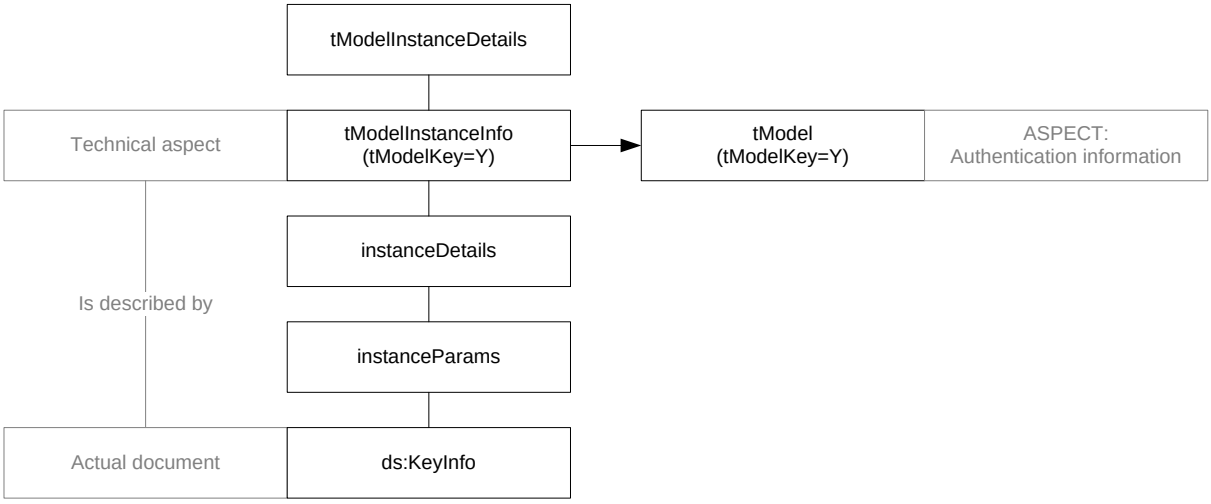

Figure 9: UDDI data model mapping (authentication information) Generic data model concepts on the left (grey), mapped to UDDI concepts (black)

6.1.4 Availability information

This section discusses the mapping of the availability information related data model.

Planned unavailability intervals

For the availability information associated with a service there data format is specified by the SIX standard. The planned unavailability interval is store in a similar fashion as the other documents. The SIX standard specifies the tModelKey that indicates the detail type.

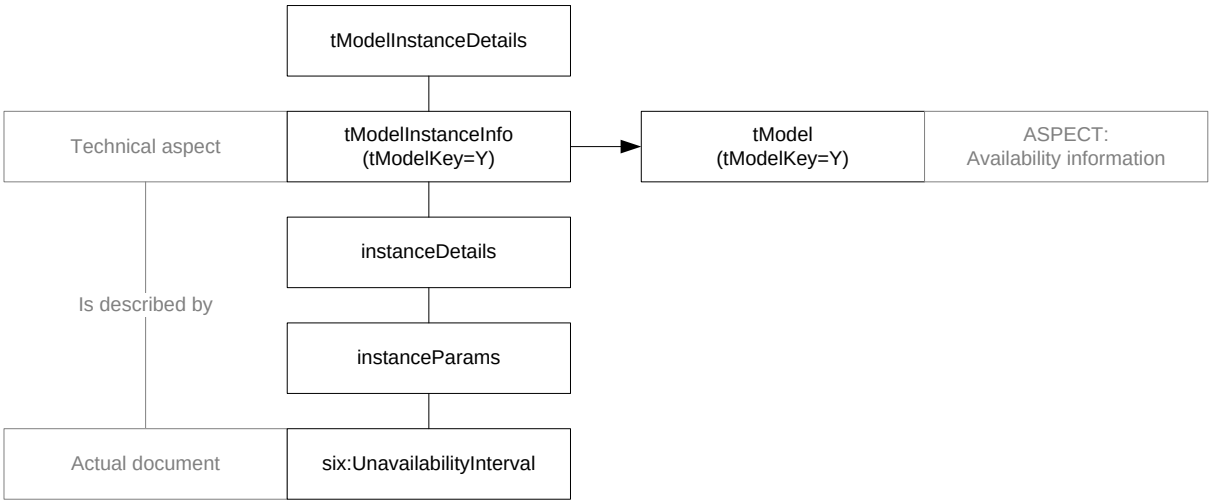

Figure 10: UDDI data model mapping (availability information) Generic data model concepts on the left (grey), mapped to UDDI concepts (black)

Current status

The current status can be stored in numerous ways, including one similar to the documents mentioned before. However because the current state consists of a single value with choose to store this value in the category bag, similar to the user type.

In the category bag of the BindingTemplate-element we add a keyed reference to a tModel representing the SIX service status. The tModel value element can contain either ONLINE or OFFLINE.

6.2 Functionality

UDDI has three relevant APIs: the publisher API, the inquiry API and the Subscription API. The publisher API provides all functions to manipulate the data in the registry. The participants and scheme organisation can use this API to manipulate the data related to the participants and the service specifications. Audit trail and lifecycle management not accommodated for by the UDDI specification.

The inquiry API provides advanced querying functionality. This API can be used by the participants to obtain information relating to a specific participant or get a copy of all data in the directory. The find_business function can be used to lookup a participant by identifier. The find_service function can be used to get a list of participants by type.

The Subscription API allows user to subscribe to information objects and be notified of updates to the information. The notification requirements can be fulfilled with this functionality.

6.3 Access control mechanism

UDDI allows any access control mechanism to be implemented but does not prescribe any mechanism or means of specifying access control policies.

The limited access control mechanism described by the specification enables public read access to all resources and allows users to create information objects and update or delete the objects it has created.

6.4 Interface compatibility

The following part of the UDDI specification:

> *UDDI registries MAY ignore the contents of SOAP header. SOAP headers that have the must_understand attribute set to true MUST be rejected with a SOAP fault - MustUnderstand. UDDI registries MAY ignore other extension headers received. (4.1.4)*

Makes it incompatible with the WS-Security and WS-addressing headers in the SIX:0208 specification, REQ15 :

> *The SIX:0208 wsse:Security element MUST have a soap:mustUnderstand attribute with the value set to '1'.*

en REQ27:

> *The SIX:0208 WS-addressing element MUST have a soap:mustUnderstand attribute with the value set to '1'.*

6.4.1 Authentication mechanisms

UDDI provides a single authentication mechanism based on username and password. This method must be supported. There is no facility to implement other authentication methods or replace the existing one. Public read access to the registry is required by the UDDI registry specification.

6.5 Conclusions

For the information above the following conclusions can be drawn:

1. The generic data model can fairly easy be mapped onto the UDDI data model. However in able to store the actual documents in the registry it is necessary to use a construction that does not conform to the intended use of that UDDI concept.
2. The functionality provided by UDDI satisfies the functional requirements.
3. The access control mechanism specified by the UDDI specification is very limited and fails to meet the requirements. However the access control mechanism is extensible. In able to meet the requirements related to access control significant effort must be undertaken to specify an additional access control mechanism. The necessity of additional specification makes implementations of the standard useless or requires also additional implementation efforts.
4. The authentication mechanism is based on username/password combinations. While other authentication mechanisms may be added, this mechanism must be implemented. The username/password based authentication mechanism provides week security.
5. The interface is not compatible with SIX:0208, since it allows for wsse:Security and WS-addressing elements to be specified in the header of SOAP messages, however requires implementations to ignore them.
6. While SIX:0208 compatibility is not strictly required, it is preferred. However the issues preventing SIX:0208 compatibility also prevent other message level security features to be incompatible.

The interface incompatibility (conclusion 5) combined with the weak security (conclusion 4) and the lack of access control mechanisms (conclusion 3) justify the further assessment of an alternative technology.

7 Proof of concept: ebXML Registry and repository service

The ebXML Registry and repository service got the second highest score in the second assessment iteration. It scored high on functionality and standard. Lower scores were achieved on use/usage. Detailed examination of the specification in conjunction with the refined requirements resulted in the following observations.

7.1 Data model mapping

This section describes the mapping of the data model of the requirements on the EBXML RS/RIM data model. The mapping is mostly of a syntactic nature, any semantic resemblance between concepts is purely coincidental.

7.1.1 Identification information

The participant is the primary object in the generic data model and as such is mapped to the ebXML RIM Organisation object, that has a similar semantic.

Participant Identifier

The participant identifier can be stored using an ExternalIdentifier object that is indended to be used to provide additional identifier information to RegistryObjects. An ExternalIdentifier references the identification scheme from which the identifier is derived. A scheme specific classification scheme for participant identifiers should be registered with the registry. This classification scheme itself should be classified as being a participant identifying classification scheme.

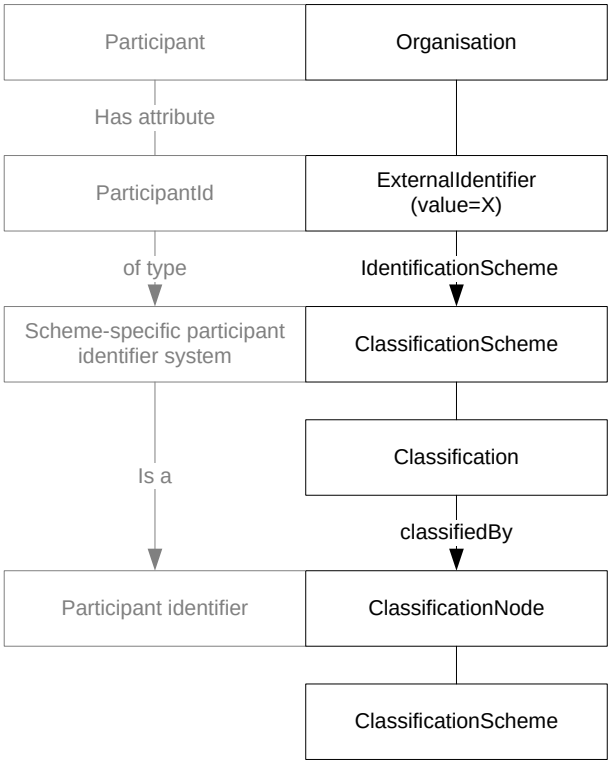

Figure 11: ebXML RIM data model mapping (participant and participant identification) Generic data model concepts on the left (grey), mapped to ebXML RIM concepts (black)

Participant contact information

To store the contact information the Organization class provides attributes for postal addresses, e-mail addresses, primary contacts, telephone numbers.

Scheme role (participant type)

The scheme role can be represented as a classified service object. For each role performed by the participant a service object can be created. The service object can be classified a representing a role of a certain type. The classification scheme used for this classification can be the scheme specific classification scheme defining the different roles in the scheme. This classification scheme should itself be classified as being a SIX participant role scheme.

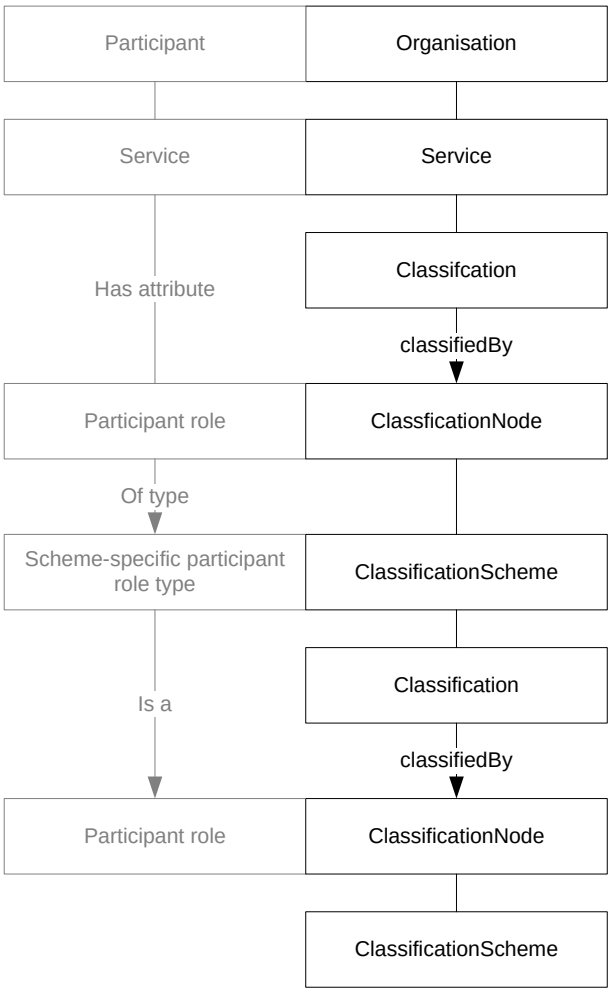

Figure 12: ebXML RIM data model mapping (service and participant role) Generic data model concepts on the left (grey), mapped to ebXML RIM concepts (black)

7.2 Addressing information

For each role performed by the participant a Service object can be registered in de directory. Each service object must have an associated Service Binding object. The service binding in turn has a specification link object associated that references the extrinsic object that holds the meta data for the repository item in which the WSDL document in

contained. The specification link should be classified as being the link that references the addressing information. The SpecificationLink object in turn references a ExtrinsicObject as specificationObject. The ExtrinsicObject contains the meta-data for the RepositoryItem that wraps the actual specification document.

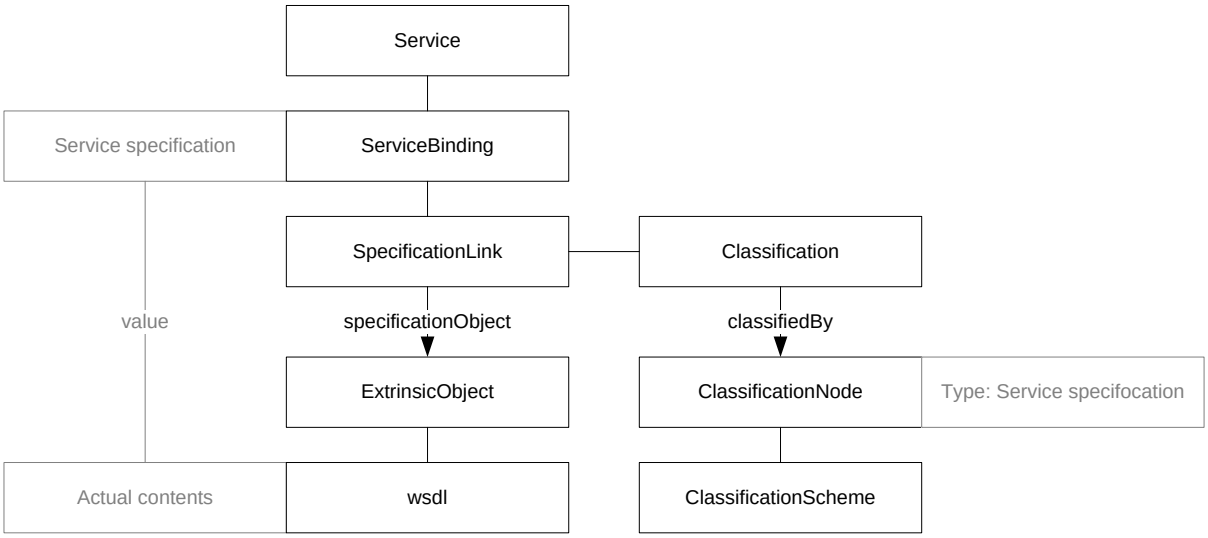

Figure 13: ebXML RIM data model mapping (addressing information) Generic data model concepts on the left (grey), mapped to ebXML RIM concepts (black)

WSDL

The actual WSDL document containing the service implementation specification can be stored in the repository wrapped in a RepositoryItem object.

7.2.1 Authentication information

Below the service binding a second specification link object can contain the certificates for the service. The specification link should be classified as being the link that references the authentication information.

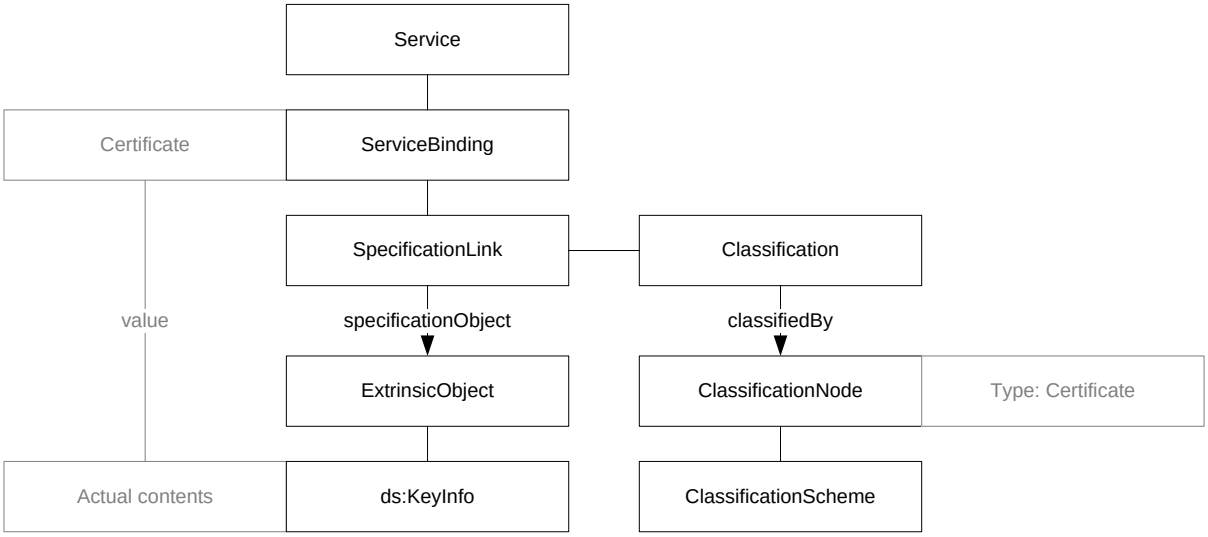

Figure 14: ebXML RIM data model mapping (authenication information) Generic data model concepts on the left (grey), mapped to ebXML RIM concepts (black)

Certificates

The actual certificates contained in a ds:KeyInfo document can be stored in a Repository item object.

7.2.2 Availability information

Below the service binding a third specification link object should contain the availability information for the service. The specification link should be classified as being the link that references the availability information.

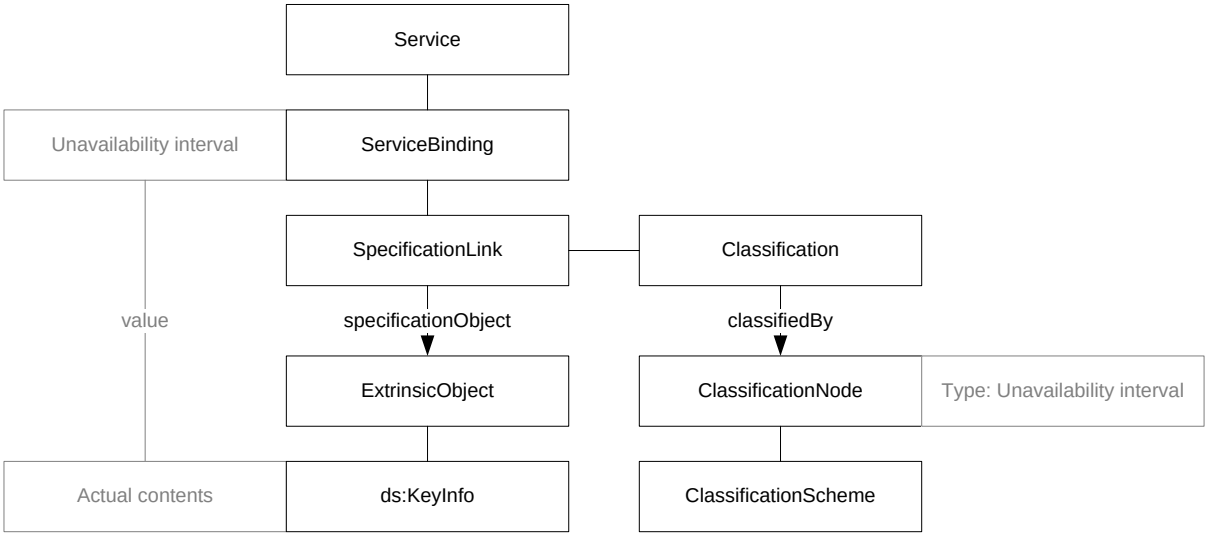

Figure 15: ebXML RIM data model mapping (availability information) Generic data model concepts on the left (grey), mapped to ebXML RIM concepts (black)

Planned unavailability intervals

The actual unavailability schedule can be stored in a Repository item object.

Current status
The current status can be stored as a classification of the Service object.

7.3 Functionality

The Lifecycle Management Protocol (2) has the basic functions to create, update and delete information contained in the directory. The EBXML specification caters for versioning and audit trail functionality.

The SubmitObjectsRequest provided by the Lifecycle management protocol accepts a set of objects to be added to the registry and/or repository. The specification can prescribe the contents of this request to create a full set of participant related objects.

The UpdateObjectsRequest provided by the Lifecycle management protocol accepts a set of objects to be updated in the registry and/or repository. The specification can prescribe the contents of this request to update the full set of participant related objects or a specific part.

The DeprecateObjectsRequest provided by the Lifecycle management protocol accepts a list of objects to be updated or a query string selecting objects to be updated. The specification can prescribe the contents of this request to update the full set of participant related objects or a specific part.

The Query Management Protocols have basic and complex functions to access the information contained in the registry and repository.

Stored queries
ebXML supports the concept of stored queries. Stored queries are predefined queries that can be reused (re-executed) without specifying the full queries but rather by referencing the previously stored query. Stored queries can be parameterized by the submitter of the stored query. The directory protocol can benefit from this functionality by defining parameterized stored queries that need to be submitted to the registry during set-up. Participants can than later leverage these queries to executed their common tasks of retrieving a single 'record' or full list.

- Retrieve participant data for single participant
- Retrieve participant data for all participants of a certain type

The filters or query language supported by ebXML RS allow both types of queries.

7.3.1 Notification mechanism

ebXML RS has a subscription based system. Users can subscribe to events, such as content updates. This mechanism can be used to implement the unavailability notification.

For this functionality the same concepts of stored queries can be used. At set-up the registry owner can submit a query that returns services with state unavailable for

unplanned (un)availability notifications and a query that returns a deprecated Extrinsic objects that represent certificates.

The EBMXL specification defines a Service NotifyAction that is taken upon event occurrence. The Service NotifyAction can deliver an event notification via a programmatic interface. Currently however the specification does not contain specifics on this feature.

7.4 Access control

ebXML RS uses the standardized XACML policy specification format. (30)

It can be used to specify access control policies that allow or prevent specific actions on resources by certain subjects. Section X already defines the policies needed for the directory protocol. These policies are expressible in XACML.

Effect	Subjects	Actions	Resources
deny	All subjects	All actions	All resources
allow	All registered users	Read	All resources
allow	All registered users	Create, Update, Deprecate, undeprecate, delete	All resources of which they have ownership
allow	Users in Scheme organizing role	All actions	All resources
allow	All registered users	Reference	All resources, expect participant Id en participant Type ClassificationNodes

7.5 Interface

Section 10.3.2.7 SOAP Message Security and HTTP/S of the ebXML RS specification states that:

When using HTTP/S between a Registry Client and a registry, SOAP message security MUST NOT be used.

This is incompatible with the SIX:0208 specification that requires both security mechanisms to be used. In the draft of version 4 of the ebXML specification this requirement is omitted.

7.6 Conclusions

From the information provided above the following conclusions can be drawn:

- The ebXML RS/RIM specification provides a lot of functionality many of which required for the solution sought in the research.

- The access control mechanism specified by ebXML RS is based on XACML which is an open standard for specifying access control policies. The mechanisms provides a standardized yet flexible implementation of access control mechanisms.
- The interface of ebXML RS/RIM is compatible with SIX:0208 with the exception of a small issue. This is the only issue where the ebXML RS specification must be overridden.

The conclusions stated above allow for the selection of ebXML RS/RIM as the basis for our directory protocol specification.

PART 4 CREATING A PROTOCOL SPECIFICATION

The technology assessment and has resulted in the selection of a technology. Based on this technology a new protocol can be specified that provides a solution to the specific problem of this research.

8 Creating the protocol specification

This section discusses the steps that were taken to draft the full specification of the directory protocol on the basis of the selected EBXML Registry and repository specifications. It discusses the structure of the specification document.

From the mapping determined in chapter 7 we derived the following need for specification. The specifications can be divided in three sections. A general section that establishes the specification as being based on the ebXML RS/RIM specifications and introduces a extra conformance profile. The second section contains the specification of the configuration of the EBXML RS implementation by the directory administrator prior to actual operation. The third section contains normative statements on the use of the directory by the different types of users.

8.1 Conformance profile

The ebXML RS specification contains two conformance profiles that specify the set of features that an implementation of the standard that is set to support a certain conformance profile must support. The two provided conformance profiles are: light and full. The 'light' conformance profile requires a minimal set of functionality while the 'full' conformance profile requires an implementation to support all functionality specified by the ebXML RS specification.

For the SIX directory protocol standard a third conformance profile is introduced. This project specifies the minimal set of functionality required to use a ebXML RS implementation as an implementation of a directory using the SIX directory protocol specification. The set of required functionality is a superset of the 'light' profile and a subset of the 'full' profile.

8.2 Configuration

The configuration related specification consists of requirements that address configuration of the ebXML RS implementation as well as requirements that define data objects that have to be submitted to the registry prior to operation.

8.2.1 Registry system configuration

The implementation configuration related requirements in the specification deal with the set up of the directory as its own authentication authority and the use of the SOAP interface binding.

The ebXML registry implementation must be configured to use itself as the authentication authority. The ebXML specification specifies that the registry may operate as its own authentication authority (section 10.4.1) but does not specify how this is to be implemented. This specification requires the implementation to operate as its own authentication authority and requires the implementation to use the contents of the directory as specified by this specification to obtain user credentials to match.

The implementation must support and use the SOAP binding as specified in ebXML RS chapter 3. This SOAP binding is compatible with the SIX:0208 Secure SOAP interface

specification that must also be implemented. The only deviation from the ebXML RS specification is that the requirement in ebXML RS section 10.3.2.7 that HTTP/S security and SOAP message security must not be used at the same time must not be used. Instead the implementation should comply with the SIX:0208 specification that states the both security mechanisms must be used in conjunction with each other.

8.2.2 Data object configuration

The data objects that need to be submitted to the registry are:

- Objects representing users;
- Classification-related objects
- The access control policy specification;
- Stored parameterized queries.

Objects representing users

The first configuration step is the creation of two required users in the registry: one user representing the participant managing role of the scheme organization and one user representing the specification managing role of the scheme organization.

The user representing the participant managing role will be the responsible user for the participant identifying objects and the scheme role classification objects (see below).

The user representing the specification managing role will be responsible user for the scheme specific service interface specification that will be stored in the registry and will own the information object classification objects (see below).

The two roles of the scheme organization are represented as two separate user to allow the access control policy to be use to limit the ability of participants to create their own identifiers and scheme-role classifications., while allowing the participants to reference the scheme-specific service interface specification.

Classification related objects

The classification model of ebXML RIM can be used for three different type of classifications in the context of this specification.

- The identification and classification of participants and role
- The classification of data object
- The representation of real-time service status

Participant identification and role classification

The ebXML mapping describes the use of the classification model to represent the participant identification and role classification information. For this representation the directory must be configured with a classification scheme for participant identification and classification information and a classification node representing a participant identification scheme and a classification node representing a participant role classification scheme.

For the particular scheme a scheme specific classification scheme for participant identification and a scheme specific classification scheme and nodes for participant role classification must be submitted to the directory.

Data object classification
A set of classification nodes with a classification scheme must be submitted to the registry. The classification nodes can be used to classify registry objects as being the object representing participant addressing, authentication and availability data.

Service status representation
A classification scheme with classification nodes representing the possible values of the current service status must be submitted to the registry.

Access control policies
The specification contains an access control policy defined in XACML. The XACML policy is parameterized. The parameters must be filled in and the resulting document must be submitted to the registry as the default access control policy for all objects.

Stored parameterized queries
The ebXML RS provide query interface may be used by participants to query the directory with any query but for convenience a set of predefined parameterized stored queries should be submitted to the registry to be used by the participant to obtain information from the registry.

A stored query for obtaining the events relevant for the real-time notification mechanism must be submitted to the registry. The query is used by the subscription and notification mechanism provided by the ebXML registry. Participants subscribed to this query are notified of changes in the result set. This query must return all changes to objects that classified with a classification node that indicates a current service status as described above.

8.3 Operation

After the directory has been set up it becomes operational. This section describes the operation related specification. It consists of both the data model mapping specification and the notification mechanism definition.

8.3.1 Data model

The data model to be included in the specification is similar to the mapping of the generic data model onto ebXML RIM as described in chapter 0. The specification prescribes this data model to be used, but it is the participant's responsibility to comply with this specification. The implementation allows any data to be and designated addressing information however will not check the contents of the submitted data for validity.

8.3.2 Notification mechanism

The real-time notification mechanism leverages the subscription mechanism provided by the ebXML registry specification. It basically consists of three parts: the query, the subscription and the notification service end-point.

A stored query that is submitted during configuration of the registry is used to retrieve all relevant events. Relevant events are those events that must trigger a notification to the participants. The query provided by the specification yield all events the involve a reference to a classification node that represents a current service status.

All participants are required to create a subscription for this stored query. The registry system will periodically execute the stored query and when changes to the result set occur send a notification to the subscribed users.

To be able to receive the notifications sent by the directory the participant must implement a service endpoint that accepts them. The notification mechanism specified in ebXML RS is used.

PART 5 CONCLUSION

The final part of the report concludes the documentation of this research. In this part the reader can find the conclusions drawn from the research activities; answering the research questions. The conclusions can be found in chapter 9. In chapter 10 the author provides some recommendations for continuing the standardization process of the SIX specifications and the further development of the specifications.

9 Conclusions

In the previous chapters the requirements for a directory protocol are discussed and refined, a generic solution model has been created, existing technologies are assessed and one is selected and the generic models and mechanisms are mapped onto the selected technology to create the directory protocol specification. In this chapter conclusions are drawn from all research activities listed above as the research questions are answered.

Assessment of the suitability of existing specifications

The first sub question addresses the process of assessing the ability of existing specification to provide a solution or be part of a solution to the problem at hand.

An assessment approach was designed and executed and resulted in the selection of existing technology that provided a solution to the problem. The approach is outlined in chapter 5. The assessment process was based on obtaining acceptable result with the time constraints of the project. While the initial approach consisted of three iterations with increasing granularity of mapping requirements to solutions if was found that the third iteration would be too time consuming to be conducted for all technologies. However it also was found that this level of detail is needed to full assess the suitability of a technology. A technology deemed the most promising in the second iteration was discarded in the third iteration.

The requirements taken from the requirements analysis document created in the preliminary research needed a refinement step to subsequently allow construction of the generic models and mechanisms.

Suitability of existing specifications

The second sub question was addressed by the technology assessment. The technology survey conducted during the preliminary research yielded a number of existing specifications that potentially solve the problem or part of the problem. These technologies were included in the assessment discussed above.

The assessment resulted in the ultimate selection of ebXML RS/RIM to base the directory protocol specification on. The ebXML RS/RIM specifications were found to be feature rich, supporting the required functionality, and with compatible interface, able to accept secure SOAP messages. The fact that it is an established OASIS UN/CEFACT standard is positive and will ease the standardization process and implementation of the new specification; however the number of existing implementation is an area of concern. Using the existing specification as a basis however reduces specification efforts and implementation efforts.

Existing specification(s) compared to creating a new specification

This question deals with the trade-off that is being made between adapting existing technology and creating new technology from the ground up. The research did not include activities that investigated the performance of a tailor-made solution in

comparison with the ebXML RS/RIM based solution. However in the assessment weights were assigned to the criteria that highly favor the use of existing standards based solutions and therefore the tailor-made solution, although with perfect scores on functional requirements, was not ranked high enough to justify in depth assessment.

Drafting of a specification
The specification was created by mapping the generic solution to the intended base specification. Details on the process are discussed in chapter 8. The intermediate construction of the generic models and mechanisms allowed for a relatively brief specification process.

Validation of the suitability of the specification
Due to time constraints this question was not answered in the research, and will need to be addressed in future research. Validation of the results is an important issue, however extremely challenging to accomplish.

How to solve the data distribution problem for multi-party networked services
The specification created as part of this research is a solution to the data distribution problem outline in the introduction of this research. The assessment of existing technology and comparison with a tailor-made solution has show this solution is a reasonably good solution. It cannot be concluded that this is the best solution as not all possible solution have been explored to the same extent.

Reflection
The e-commerce environment is highly volatile. The current results are based on the current situation, however it may be expected that the environment changes a lot in the near future. The SIX specifications will be competing in the marketplace with many other initiatives.

The weights of the multi-criteria analysis of the second iteration of the assessment heavily favor existing standards. While this may be preferable the field of e-commerce communication standards of which these directory protocol specifications are part off is very much in development. The value of an existing established standard should be reevaluated as currently many competing standards occur all hoping to become the defacto standard while not one single has emerged.

The second iteration of the assessment resulted in a ordered list of technologies starting with the technology most likely to be able to provide a good solution to the problem. UDDI was at the start of this list and therefore selected to enter the third iteration of the assessment process. However in this iteration with the increased level of detail en depth of the assessment issue were uncovered that resulted in the reconsideration of UDDI as the best potential solution and justified the additional effort to also subject ebXML (the technology ranking second in the second iteration results) which in turn did proof to provide a good solution. At this point the question is justified whether or not the assessment process design was done right.

The specification has not yet been field tested and the theoretic advantages of working with an established technology have not yet been asserted in the real-world.

Results are heavily based of knowledge and experience from inside Innopay, while experience is extensive and knowledge of high quality it might represent an single-side view and a bias towards a certain solutions or approach.

10 Recommendations

The research for this project has been conducted within the time-restrictions of the project, during the execution of the research activities a number of issues has come up that have not been fully explored. Furthermore the nature of the project is such that it cannot be regarded as a closed-ended research, while this particular project has completed its results will needed to be subject of constant (re)evaluation as the environment changes. In this section a number of recommendations for development and use of the results is set forth.

10.1 Design and organize standardization process and organization

Innopay aims to elevate the SIX specifications to a generally recognized standard, similar to the standard it is based on, such as WS-addressing and ebXML. With this objective in mind the specification that was drafted in this research has an extensive standardization process ahead. For the objective to be reached this standardization process should be designed carefully and strictly managed.

To create more traction it might be wise to establish a new independent organization to maintain the specifications and govern the development and standardization process. Alternatively the specification might be transferred to an existing of newly establish committee at an established standardization institute, such as OASIS or W3C.

Independent of the chosen form this organization should issue a public Request for Comment to allow academic and industry specialist to provide input on the draft specification. At the same time the organization should actively proceed to approach industry specialists for feedback and push for implementation of the specification in pilot or production systems. The involvement of these professionals is imported to created the necessary basis of support for the formal standardization process as well as for the opportunity to field test.

10.2 Improve robustness of the SIX specifications

In able to make the SIX specification a more stable force in the market and competition against other emerging specifications, as well as to affirm the internal structure of the specification it is a good idea to more formally record the principals on which the SIX specifications are created. It is recommended that, to replace the current informally described requirements, an additional specification is added that captures the fundamental principles that together set the context for which the SIX specifications are applicable and the general requirements imposed by this context.

Because of the volatile environment and likelihood of emerging new technologies it is recommended to introduce the requirements and generic solution model as a companion specification in the standards suite. The ebXML RS/RIM based specification should than be considered as a possible, and at this point mandatory, implementation of this meta specification. This allows for quick specification of new implementations based on the meta-standard and an alternative technology.

The specification process will then be similar to the process conducted in the research that involved mapping the generic solution models and mechanisms to the models and mechanism exposed and provided by the technology. The use of the meta-specification eliminates the need to revisit the principles that are behind the directory specification and allow the developer to focus on the implementation specific issues.

10.3 Monitor market and technology developments

As the specification drafted in this research is the first version of the document and it is going to exist in a highly volatile environment in which many new developments are to be expected, it is essential to monitor these developments. More specifically the following is recommended:

Monitor comparable initiatives

SIX is not the only initiative to provide a standardized generic infrastructure for online real-time e-commerce transactions. Other initiatives are under development with similar or wider scope. The progression on development of these initiatives should be monitored closely and depending on the findings step should be taken.

Monitor ebXML specification development

The use of ebXML RS/RIM as a basis for this specification creates a dependency. The monitoring of the development of the base specification is therefore very important. As this specification is based on version 3.0 of the ebXML specification, which is obviously fixed; a forth version is already under development. In de specification we have seen already one instance in which an improvement in version 4 eliminates an additional requirement. Publication of a new version of ebXML should be sufficient cause for investigating the migration of the SIX standard to version 4 of ebXML. Besides monitoring the development Innopay could also choose to directly influence the drafting of ebXML specification by joining the OASIS technical committee that governs the drafting process.

Monitor use and existence of implementations of ebXML (RS/RIM)

The current and future use of the ebXML RS/RIM specification is and should be an issue of concern and should be monitored closely. If the ebXML specification fails to gain popularity in the coming years it would be unwise to continue basing the specification on it, and reassessing the potential technologies should be considered.

Monitor development of alternative solutions

The current draft specification is based on the ebXML RS/RIM specifications. The technology assessment and selection process in this research resulted in the selection of this specification to function as a bases. The assessment criteria used in the process includes criteria of which the scoring might change over time. ‘Soft’ criteria such as: standardization and use/usage. Additionally the assessment only included technologies that were discovered during the technology survey conducted in the preliminary research project. Both the ebXML standards and the SIX Standards are not the only initiatives in development in their respective fields. The emergence of new applicable technologies may be relevant and change the outcome of the assessment.

11 Bibliography

1. **Innopay BV.** *SIX Standards - General introduction.* [Powerpoint presentation] Schiphol : s.n., 2008.

2. **Fuger, Sally, Najmi, Farrukh and Stojanovic, Nikola.** ebXML Registry Services and Protocols Version 3.0. *OASIS-open.org.* [Online] 05 02, 2005. [Cited: 07 27, 2009.] http://docs.oasis-open.org/regrep-rs/v3.0/.

3. **Van Eijnatten, Ruben.** *Preliminary requirements analysis and technology survey SIX:2503 Directory protocol.* Delft : s.n., 2009.

4. **ABN AMRO.** What is iDEAL? *ABN AMRO .* [Online] [Cited: 07 27, 2009.] http://www.transactionbanking.abnamro.com/attachments/regions/iDEAL.pdf.

5. **Bottelberghs, Leendert, et al.** *SIX:0208 Secure SOAP Interface.* Schiphol-Airport, The Netherlands : s.n., 01 13, 2009.

6. **Clement, Luc, et al.** UDDI.org. *UDDI Version 3.0.2.* [Online] 10 19, 2004. [Cited: 07 27, 2009.] http://uddi.org/pubs/uddi-v3.0.2-20041019.htm.

7. *Network externalities, competition and compatibility.* **Katz, Michael L. and Shapiro, Carl.** 3, 1985, The American Economic Review, Vol. 75, pp. 424-440.

8. *Two-Sided Markets: An overview.* **Rochet, Jean-Charles, and Jean Tirole.** s.l. : mimeo, 2004.

9. *eBay's Paypal: Balancing Marketplace And Regulatory Regimes.* **Selby, John and Manning, Christopher J.** 6, 2008, Vols. Computer Law Review International,.

10. **Eisenmann, Thomas, Geoffrey Parker, and Marshall W. Van Alstyne.** Strategies for Two-Sided Markets. *Harvard Business Review.* 10 2006.

11. **Chakravorti, Sujit and Roson, Roberto.** *Platform Competition in Two-Sided Markets: The Case of Payment Networks.* Chicago : Federal Reserve Bank of Chicago, 2004.

12. **Kalbfleisch, Pieter.** NMA - Speech Pieter Kalbfleisch: 'Consumer welfare, innovation and competition' - Innsbruck, 26 februari 2009. *NMA.* [Online] 02 26, 2009. [Cited: 03 19, 2009.] http://www.nmanet.nl/nederlands/home/Actueel/Publicaties/Lezingen_en_speeches/Speech_Pieter_Kalbfleisch_Consumer_welfare_innovation_and_competition_Innsbruck.asp.

13. **Fielding, Roy Thomas.** *Architectural Styles and the Design of Network-based Software Architectures.* Irvine : University of California, 2000.

14. *RFC 793: Transmission control protocol.* **Postel, J.** 1981, ARPANET Working Group Requests for Comments.

15. *RFC791: Internet Protocol.* **Postel, J.** 1981, ARPANET Working Group Requests for Comments.

16. **Fielding, Roy, et al.** Hypertext Transfer Protocol - HTTP/1.1. *W3.org.* [Online] 06 1999. [Cited: 07 27, 2009.] http://www.w3.org/Protocols/rfc2616/rfc2616.html.

17. **Klensin, J.** RFC2821: Simple Mail Transfer Protocol. [Online] http://www.ietf.org/rfc/rfc2821.txt.

18. **J. Postel, J. Reynolds.** RFC 959: File transfer protocol. [Online] http://www.w3.org/Protocols/rfc959/.

19. *Analysis of the SSL 3.0 protocol.* **D Wagner, B Schneier.** Oakland : USENIX Association., 1996, Vols. Proceedings of the 2nd USENIX Workshop on Electronic Commerce (EC-96).

20. **Dierks, T and Rescorla, E.** RFC 5246 The Transport Layer Security (TLS) Protocol. *The Internet Engineering Task Force.* [Online] 08 2008. [Cited: 04 11, 2009.] http://tools.ietf.org/html/rfc5246.

21. **Gudgin, M. and Hadley, M. and Mendelsohn, N. and Moreau, J.J. and Nielsen, H.F.** SOAP version 1.2 part 1: Messaging framework. *World Wide Web Consortium (W3C).* [Online] 04 27, 2007. [Cited: 04 11, 2009.] http://www.w3.org/TR/soap12-part1/.

22. **Christensen, Erik, et al.** Web Services Description Language (WSDL) 1.1. *W3C.org.* [Online] 03 15, 2001. [Cited: 07 27, 2009.] http://www.w3.org/TR/2001/NOTE-wsdl-20010315.

23. **OASIS.** *Web Services Security: SOAP Message Security 1.0 (WS-Security 2004).* 2004.

24. **Iwasa, Kazunori.** Web Services Reliable Messaging TC WS-Reliability 1.1. [Online] http://docs.oasis-open.org/wsrm/ws-reliability/v1.1/wsrm-ws_reliability-1.1-spec-os.pdf.

25. **Box, D. and Curbera, F. and others.** Web Services Addressing (WS-Addressing). *The World Wide Web Consortium.* [Online] 08 10, 2004. [Cited: 04 11, 2009.] http://www.w3.org/Submission/ws-addressing/.

26. **WS-I Organization.** Welcome to the WS-I Organization's Web site. *WS-I Organization.* [Online] WS-I Organization. [Cited: 04 11, 2009.] http://www.ws-i.org.

27. **Web Services Interoperability Organization.** Deliverables - Overview. *Web Services Interoperability Organization.* [Online] Web Services Interoperability Organization. [Cited: 04 11, 2009.] http://www.ws-i.org/deliverables.

28. **EPC.** e-Mandates e-Operating Model - Detailed Specification. *European Paymets Counsil.* [Online] 04 21, 2009. [Cited: 07 27, 2009.]

http://www.europeanpaymentscouncil.eu/knowledge_bank_detail.cfm?documents_id=224.

29. **Larmouth, John.** The use of ASN.1 in the specification of e-business standards. *ASN.1 Consortium.* [Online] 10 2003. [Cited: 03 24, 2009.] http://www.asn1.org/paper/Seoul%20ASN.1%20presentation.ppt.

30. **Godik, Simon.** OASIS eXtensible Access Control Markup Language (XACML) Version 1.0. [Online] http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-01.pdf.

31. **Nadalin, Anthony.** Web Services Security: SOAP Message Security 1.0. [Online] http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf.

32. **Eastlake, Donald.** XML-Signature Syntax and Processing. [Online] http://www.w3.org/TR/2001/PR-xmldsig-core-20010820/.

33. **D Ferraiolo, J Cugini, DR Kuhn.** Role-based access control (RBAC): Features and motivations. [Online] http://brutus.ncsl.nist.gov/groups/SNS/rbac/documents/ferraiolo-cugini-kuhn-95.pdf.

34. **Fuger, Sally, Najmi, Farrukh and Stojanovic, Nikola.** ebXML Registry Information Model Version 3.0. [Online] http://www.oasis-open.org/committees/regrep/documents/3.0/specs/regrep-rim-3.0-cs-01.pdf.

35. **Sermersheim, J.** RFC4511: Lightweight Directory Access Protocol (LDAP). [Online] http://tools.ietf.org/html/rfc4511.

36. **Organization for the Advancement of Structured Information Standards.** Directory Services Markup Language v2.0. [Online] 04 30, 2002. [Cited: 07 27, 2009.] http://www.oasis-open.org/committees/dsml/docs/DSMLv2.doc.

37. **Bray, T. and Paoli, J. and Sperberg-McQueen, C.M. and Maler, E. and Yergeau, F. and Cowan, J.** Extensible Markup Language (XML) 1.1 (Second Edition). *World Wide Web Consortium (W3C).* [Online] 09 29, 2006. [Cited: 04 11, 2009.] http://www.w3.org/TR/xml11/.

38. **MuleSoft.** MuleGalaxy. [Online] http://www.mulesoft.org/display/GALAXY/Home.

39. **WSO2.** WSO2 Governance Registry. [Online] http://wso2.com/products/governance-registry/.

40. **wikipedia.org.** Representational State Transfer. [Online] http://en.wikipedia.org/wiki/Representational_State_Transfer.

41. **Ford, Warwick and Hallam-Baker, Philip.** XML Key Management Specification (XKMS). *W3C.org.* [Online] 03 30, 2001. [Cited: 07 27, 2009.] http://www.w3.org/TR/2001/NOTE-xkms-20010330/.

42. **Bradner, S.** Key words for use in RFCs to Indicate Requirement Levels. *IETF.org.* [Online] March 1997. [Cited: 07 27, 2009.] http://www.ietf.org/rfc/rfc2119.txt.

43. **Fallside, David C. and Walmsley, Priscilla.** XML Schema Part 0: Primer Second Edition. *W3C.org.* [Online] 10 28, 2004. [Cited: 07 27, 2009.] http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/.

44. **Chadwick, D.** *Understanding X. 500: The Directory.* London : Chapman & Hall, Ltd., 2004.

45. **Innopay BV.** Innopay - Payment consultants - About us. *Innopay - Payment consultants.* [Online] [Cited: 03 01, 2009.] http://www.innopay.com/index.php/plain/about_us.

PART 6 APPENDICES

12 Appendix A: Requirements overview

Overview of all requirements in (3)

The keywords “MUST”, ”MUST NOT”, ”REQUIRED”, ”SHALL”, ”SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY” and ”OPTIONAL” in this document are to be interpreted as described in (42)

REQ1 The directory service specified by the standard MUST retain identification information.

REQ2 The participants in the network communicate with each other using SIX:0208 Secure SOAP Interface, the identification information stored MUST support identification information needed by this protocol.

REQ3 The directory service specified by the standard MUST retain authentication information.

REQ4 The participants in the network communicate with each other using SIX:0208 Secure SOAP Interface, the authentication information stored MUST support authentication information needed by this protocol.

REQ5 The directory service specified by the standard MUST retain addressing information.

REQ6 The participants in the network communicate with each other using SIX:0208 Secure SOAP Interface (5) the addressability information stored MUST support addressing information needed by this protocol.

REQ7 The directory service specified by the standard MUST retain availability information.

REQ8 The directory service specified by the standard MUST retain the service interface specifications.

REQ9 The directory service specified by the standard MUST have a validity period specified for each information item.

REQ10 The directory service specified by the standard MEST retain an audit trail of each record.

REQ11 The directory service specified by the standard SHOULD provide the ability for a participant to retrieve the information set for a single participant with the participants identifier.

REQ12 The directory service specified by the standard SHOULD provide the ability for a participants to retrieve the identification information for all participants of a specific participant type.

REQ13 The directory service specified by the standard SHOULD provide the ability for a participant to retrieve the full contents of the directory in a single operation.

REQ14 The directory service specified by the standard MUST provide at least one of the query methods specified in REQ11 and REQ13.

REQ15 The directory service specified by the standard MUST provide the ability for a participant to update its entry in the directory.

REQ16 The directory service specified by the standard MUST provide the ability for a participant to notify / be notified in real-time of important events.

REQ17 The directory service specified by the standard MUST provide the ability for the scheme organization to manage service interface specifications

REQ18 The directory service specified by the standard MUST provide the ability for the scheme organization to add a new participant to the directory

REQ19 The directory service specified by the standard MUST provide the ability for the scheme organization to remove a participant from the directory

REQ20 The directory service specified by the standard MUST provide the ability for the scheme organization to update specific data related to a participant

REQ21 Read access to all information MUST be restricted to participants

REQ22 Update operations on information, both participant information and service specifications, MUST be restricted to the owner of the information, either the participant and/or the scheme organization.

REQ23 Create operations MUST be restricted to the scheme organization.

REQ24 Delete operations MUST be restricted to the scheme organization.

REQ25 The protocol specified in this standard MUST support measures to ensure integrity, authenticity and non-reputability of interactions with the directory.

REQ26 The protocol specified in this standard MUST support measures to ensure confidentiality of interactions with the directory.

REQ27 For point-to-point communication between the participants and directory the standard SHOULD specify/require the SIX:0208 Secure SOAP Interface to be used.

REQ28 Message format (specified by the standard) SHOULD be an XML-based format expressible in XML Schema (XSD) (43)

REQ29 Existing standardized message and/or data formats SHOULD be reused;

REQ30 The standard SHOULD, if possible, be based on existing standards

REQ31 If possible, off-the-shelf software components SHOULD be able to provide an implementation of the standard.

REQ32 The degree of integration of the parts of the standard should be high.

REQ33 The standard MUST fit with other sub-standards in SIX Standards.

13 Appendix B: Requirements refinement

The appendix contains the results of the requirements refinement process discussed in section 3.

The keywords “MUST”, ”MUST NOT”, ”REQUIRED”, ”SHALL”, ”SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY” and ”OPTIONAL” in this document are to be interpreted as described in (42)

13.1 Areas of concern

The requirements are structured akin to the following areas of concern:

- Data model
- Notifications
- Interface

13.2 Data model

The information model is refined with

13.2.1 Identification information

High level requirements

The directory service specified by the standard MUST retain identification information. (REQ1)

Low-level requirements

The identification information retained by the directory MUST consist of:

REQ1 A unique (at least within the scope of the network) party identifier

REQ2 One or more values indicating the role the participant plays in the network

A basic set of contact information consisting of at least:

REQ3 Contact person name

REQ4 E-mail address

REQ5 Phone number

13.2.2 Authentication information

High level requirements

The directory service specified by the standard MUST retain authentication information.

The participants in the network communicate with each other using SIX:0208 Secure SOAP Interface, the authentication information stored MUST support authentication information needed by this protocol.

Low-level requirements
The directory service specified by the standard MUST retain the following authentication information:

REQ6 Multiple X.509 certificates

REQ7 A validity period for the certificate

REQ8 A status indication for the certificate, indicating certificate revocation

13.2.3 Addressing information

High level requirements
The directory service specified by the standard MUST retain addressing information.

The participants in the network communicate with each other using SIX:0208 Secure SOAP Interface the addressability information stored MUST support addressing information needed by this protocol.

13.3 Low-level requirements
The directory service specified by the standard MUST retain the following addressing information:

REQ9 Multiple WSDL-documents specifying service end-points

REQ10 References to service specifications in the directory

13.3.1 Availability information

High level requirements
The directory service specified by the standard MUST retain availability information.

Low-level requirements
The directory service specified by the standard MUST retain the following availability information:

REQ11 Per service, a field indicating the current status of the service. Available or unavailable.

REQ12 Per service, a list of planned unavailability periods, indicating expected begin and end time

13.3.2 Service specifications

High level requirements
The directory service specified by the standard MUST retain the service interface specifications.

Low-level requirements
The directory service specified by the standard MUST retain the following service specifications:

REQ13 On a participant independent level, multiple WSDL documents

REQ14 On a participant independent level, multiple WS-SecurityPolicy documents

13.3.3 Validity period and audit trails

High level requirements

The directory service specified by the standard MUST have a validity period specified for each information item.

The directory service specified by the standard MUST retain an audit trail of each record.

Low-level requirements

The directory service specified by the standard MUST retain the following information:

REQ15 For each information item: begin and end dates for of the validity of the data

REQ16 Updates to information items must be logged

13.4 Notifications

The following requirements are related to notification isssues.

The directory service specified by the standard MUST provide the ability for a participant to notify / be notified in real-time of important events.

The rationale behind this requirement is based on the prevention of a degraded end-user experience. The degraded user experience occurs when end-users have to wait while the participant attempts to connect to an unavailable party or needs to query the directory for updates. Also offering an end-user the option to select a party that is not available at that time needs to be prevented. Have a notification mechanism prevents this situation by providing the ability for participants to update their local data as soon as an update is available and also to be actively notified of unscheduled unavailability.

Hence, the following notification events are indentified:

REQ17 Unscheduled (un)availability

REQ18 Certificate revocation

Method of notification

REQ19 The scalability requirement implies that the notification MUST be machine readable so that participant systems can automatically take appropriate action upon receiving a notification.

13.5 Interface

The following four high-level requirements relate to the interface.

The protocol specified in this standard MUST support measures to ensure confidentiality of interactions with the directory. (REQ25)

The protocol specified in this standard MUST support measures to ensure integrity, authenticity and non-reputability of interactions with the directory. (REQ26)

For point-to-point communication between the participants and directory the standard SHOULD specify/require the SIX:0208 Secure SOAP Interface to be used. (REQ27)

Message format (specified by the standard) SHOULD be an XML-based format expressible in XML Schema (XSD) (REQ28)

Since the SIX:0208 standard ensures confidentiality and has measures to ensure integrity authenticity and non-reputability, the low-level requirements can be taken from the SIX:0208 specification.

14 Appendix C: Assessment results

In this section the results of the technology assessment are discussed. The first section contains the results of the first interaction iteration; the second section addresses the results of the second iteration.

14.1 First iteration results

This section provides the results of the first assessment iteration. For each technology the scores on the criteria: standardization, use and usage and functionality.

The technologies included in this iteration are:

- Lightweight Directory Access Protocol (LDAP)
- XML Key Management Specification (XKMS)
- Universal Description Discovery and Integration (UDDI)
- ebXML Registry/repository (ebXML RS/RIM)
- MuleGalaxy
- WSO2 Governance registry

Lightweight Directory Access Protocol (LDAP)

The LDAP specification (35) defines a general-purpose directory system and service that can store and retrieve any type of data in/from its tree-based data structure. LDAP is not specifically designed for service specification registry but can store any type of data. We consider version 3 of the LDAP specification. Most modern implementations of LDAP include a DSML interface (36) that allows XML formatted messages for interacting with the LDAP service. We consider these standards in conjunction with each other.

Standardisation

X.500 (44) is a series of standards covering directory services. It was developed by ITU-T in the early 1990s. ISO was a partner in developing the standards and the standard was incorporated into the Open Systems Interconnection suite of protocols. The original X.500 protocols use the OSI networking stack.

The complex OSI networking stack however is rarely used in practice and has been replaced by the more popular and simpler TCP/IP stack and for this stack an alternative, simplified implementation of the X.500 Directory Access Protocol (DAP) was developed by the Internet Engineering TaskForce (IETF): Lightweight Directory Access Protocol.

The third version of the LDAP specification has been published in June 2006 in IETF RFCs 4510 up to 4519. The LDAP standard is very mature and supported by major organisations and many software vendors.

Use and usage

LDAP is popular and widely used. All major software vendors have implementations that are based on the LDAP specifications. Popular implementations include: ActiveDirectory by Microsoft, Novell eDirectory, Sun Directory Server, Oracle Internet Directory, Apache

Directory Server and OpenLDAP Server. With extensive use of many implementations and a large variety of usage LDAP has a very strong position.

LDAP based services are used for various purposes and in a variety of environments. The most common use is to facilitate enterprise single sign on, but use ranges from storage of user data for a small website to storage of account data for millions of mobile phone subscribers of a mobile network operator.

Functionality
The LDAP server can store any data in its tree-based data structure, both meta-data and documents can be handled.

Notification support, sometimes referred to as synchronisation support, is not provided by the core specifications but there exists and RFC that specifies it, LDAP Client Update Protocol (LCUP), however none of the major implementation has included this. Many implementation implement similar functionality in an implementation-specific fashion.

XML Key Management Specification (XKMS)
The XKMS specification (41) defines a protocol for interacting with a key store. The message format of the protocol is XML-based. We consider XKMS version 2.0.

Standardisation
The standard specifies an XML-based protocol for distributing and registering public keys or certificates. XKMS standard was developed in a joint effort by Microsoft, Verisign and WebMethod under the auspicien of the W3C, a leading standardisation organisation. The second version of the XML Key Management Specification is promoted to W3C Recommendation on June 28, 2005.

Use and usage
There is a small number of implementations including one by Oracle as part of the Oracle XML Security toolkit and an open-source implementation called OpenXKMS. The latest release of OpenXKMS was on March 2nd, 2009, on April 23rd 2009 it was downloaded 25 times, total number of downloads for all versions of the server package did not exceed 1000. From these figures the use of OpenXKMS appears to be limited. The list of implementation on the XKMS working group website count 6 XKMS 2.0 implementations and 4 XKMS 1.0 implementations, of which only 2 links are still active. The Oracle implementation is the most noteworthy one. While the use of XKMS is difficult to determine the various indicators point to the fact that it is limited. There exist few implementations and has been no development since the release of the standard in 2005.

Functionality
The XMKS standard consists of two major parts: the XML Key Information Service Specification (X-KISS) and the XML Key Registration Service Specification (X-KRSS). X-KISS allows a client to delegate part or all of the tasks required to process XML Signature <ds:KeyInfo> elements to an XKMS service. Its use will reduce the complexity of applications using public key infrastructure for creating trust. X-KRSS describes a

protocol for registration and management of public keys. It allows for the registration of additional identification information with the public keys. Keys may be generated by the users and uploaded to the service of be generated by the service and included in the result. The protocol supports the basic functions: register, reissue, revoke and recover to manipulate the registry.

There are no provisions for bulk operations or data synchronization in XKMS but there exists a W3C Working draft specification called X-Bulk that adds bulk operations to the standard. The specification however has no provisions for data synchronization.

The data model of the XKMS specification only supports authentication information which is insufficient to base the directory protocol standard on. XKMS can however be considered in combination with a solution based on other technology.

Universal Description Discovery and Integration (UDDI)
The UDDI registry specification (6) was designed to provide a means for storing web service specification meta-data. Together with the SOAP and WSDL specification it forms the core of the original web services specifications stack. Version 4 is currently under development but we consider version 3.0 of the specification is this assessment.

Standardization
Together with WSDL and SOAP UDDI forms the core set of web services technologies from the OASIS organization that should enable service oriented architectures on the web. It was originally designed to provide a global public directory in which business can list the services their offer that could be queried by anyone looking for specific online services.

It is supported by major software vendors such as IBM en Oracle. The third version of the UDDI specification was finalized in 2004.

Use and usage
Despite the existence of some public directories the most common use of UDDI is on private enterprise networks. UDDI server implementations are available from all major vendors, including Microsoft, Novell, Sun. There is also an open source implementation called OpenUDDI. UDDI is supported by most major software vendors, with the exception of SUN that actively supports ebXML Registry as an alternative to UDDI.

UDDI has many implementation of most major software vendors. Despite its intended use as public web service registry it is mostly used on internal networks.

Functionality
A UDDI registry contains meta-data related to web service providers and the provided services. The actual documents describing the services and such cannot be stored in the registry. Version 3 of the UDDI specification adds support for notifications to the protocol.

ebXML Registry/repository (ebXML RS)
The ebXML RS specification (2) is accompanied by the ebXML Registry information model (ebXML RIM) specification that defines information model used by compliant registries. Besides these registry related specification the ebXML suite, designed to enable a global electronic market, includes a messaging service specification, a collaboration protocol specification and a business process specification. For this assessment we consider version 3.0 of the ebXML specification.

Standardization
In joint effort of the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) and OASIS the e-business XML (ebXML) set of standards was developed. ebXML aims to provide a set of specifications that enables electronic trading relationships between business partners.

With the backing of the UN and OASIS this standard is very promising. While it is currently in its third version, completed in 2005, and work on a forth is in progress it is still under active development.

Use and usage
Use of this standard appears to be limited. There is one implementations of the standard from Sun Microsystems that has been donated to the open-source community and is the reference implementation of ebXML RS. It is called: freebXML. There are also implementations that are used in a limited number of pilot projects mostly in government and automotive industry.

Functionality
The registry and repository solution specification, which was added in version 3.0, was developed with similar goals as the SIX directory protocol and provides a reasonable fit. It supports both meta-data and documents. Notifications are also supported.

MuleGalaxy
Standardisation
MuleGalaxy (38) is a product of MuleSource and as such not based on a standard. Version 1.5 was released early 2009. The software is provided as an open-source package.

Use and usage
There is no publicly available data on the use and usage of this product.

Functionality
This 'Service oriented architecture governance platform' is a SOA registry and repository that allows for storage of any artefact and has the ability to extract meta-data from these artefacts to populate the registry. Content validation and content lifecycle management are supported. The application exposes its data through a graphical user interfaces as well as a REST-style web service interface that is not based on an open standard. The repository is searchable with a proprietary query language.

There is a good functional fit between the requirements of the SIX standard and the features of this product. However the fact that this product does not use a widely recognized standard for its interfaces will seriously impact ease of adoption in a negative way.

WSO2 Governance registry

Standardization

This 'governance registry' is part of the open-source WSO2 SOA middleware solution (39) created by a company with the same name. The latest version, version 3, was released in 2009.

Use and usage

There is no publicly available data on the use and usage of this product.

Functionality

It can maintain a library of services and related specifications. It support role-based access control and live cycle management and offers user the ability to provide feedback on service specification. It has a web-based graphical user interfaces as well as REST-style web services to access and manipulate the contents of the registry. The web service interface is not based on an open standard.

There is a good functional fit between the requirements of the SIX standard and the features of this product. However the fact that this product does not use a widely recognized standard for its interfaces will seriously impact ease of adoption in a negative way.

14.2 Second iteration results

This section discusses the result of the second assessment iteration. During this iteration the technologies have been assessed on the following criteria: data model, functionality, access control mechanism and interface compatibility.

The result of the assessment per technology are stated below:

UDDI

Data model

Each business entity contains descriptive information about a business or organization and information about the services that it offers. The business entity information contained in the directory includes a name, description and contact information.

Each business entity has a number of business service entries in the register. Each of these entries contains descriptive information in business terms and bindings to technical descriptions of the service.

UDDI in itself does not provide specific definitions of technical descriptions but offers an extensible model for it, called tModels. There is a standard tModel specification for WSDL

documents, and additional tModel can be specified for other information, such as the authentication and availability information.

The tModels provided references (URI/URL) to the technical specifications but the UDDI registry does not contain the actual specification data itself. The data needs to be exposed through other means.

Since the design goals of UDDI are very similar to those of the SIX directory protocol, that data model of UDDI is very likely to fit with the requirements and will require moderate specification efforts.

Functionality
At its core UDDI provides a directory of registry of business entities with basic business related information about these entities that provide business services. While UDDI certainly allows retrieval of data based on a single identifier part of the power of UDDI is in its ability to process more complex queries, functionality that is not used in the context of SIX. UDDI specification also includes mechanisms for replication of the data set. The latest version also has support for a subscription service that allows user to be notified of changes to the registry that are included in a certain query.

All simple operations are supported and also more advanced querying is possible. With the third version of the specification a replication and federation mechanism specification was added. However these features only apply to the registry, for the repository extra specification is needed.

Audit trail and lifecycle management features are not included in the specification, but a subscription model based notification mechanism was also added in this version.

Access control mechanisms
Version 3.0.x of the UDDI specification defines the concept of information ownership. However access control and authorization are complete left up to the implementation.

Interface
UDDI uses XML formatted messages and a SOAP HTTP binding. However the UDDI SOAP binding is not compatible with the SIX:0208 specification.

LDAP
Data model
LDAP, like X.500 DAP, uses a tree structure to store its directory entries. Each entry is a set of attributes that consist of an attribute name, attribute description and one or more attribute values. The structure of entries (objects) and attributes is defined in a schema that specifies object classes and attribute types. Each entry has a unique identifier called the distinguished name, consisting of its relative name and the name of its parent.

LDAP uses binary data format, ASN.1, and there is a LDAP Data Interchange Format (RFC 2849) for representing the data that is text based. An LDAP directory can store any data.

There are a number of RFC defining data models for specific data including X.509 certificates (IETF RFC 4523) and a memo describing a schema for UDDI data (IETF RFC 4403).

Functionality
LDAP specifies 5 basic operations: search, compare, update, bind and unbind. Bind and unbind, respectively start and stop an authenticated session with the LDAP server. Search and compare allow the client to retrieve data or to check attribute values. Update allows the client to update an entry in the directory. These operations fulfil the individual record operation requirements.

Synchronization is not a part of the LDAP specification but there exists a memo describing this functionality, RFC4533, and another approach is described in memo RFC3928. These specifications are implemented by a limited number of directory server vendors. These memos also specify a method for real-time distribution of data.

Audit trails are not maintained by default. There is no notification support.

Access control mechanisms
There is no access control mechanism specified in the LDAP v3 specification. The informational IETF RFC 2820 specifies requirements for such a mechanism, but no such mechanism has been standardized. Various implementations of LDAP server however have access control mechanisms implemented.

Interface
LDAP is an application level protocol on top of the TCP/IP stack. The data exchange is binary, and as such by default not SIX:0208 compliant as it is neither XML formatted or SOAP enveloped.

However there exists an OASIS standard called Directory Service Markup Language (DSML) that specifies XML formatting for LDAP entries and request and response messages. The DSML standard also provides a SOAP binding using HTTP/1.1. But this DSML only transforms the format of the messages and does not enable LDAP to 'understand' XML. The interface specification would become complicated.

ebXML Registry and repository
Data model
Both the registry for meta-data and repository for actual artefacts have a very extensible data model.

Custom object classes and relationship types can be defined, but the specification of the ebXML Registry information model (ebXML RIM) contains a pre-defined data model for identification, classification, provenance, service information, event information and access control information.

Unlike UDDI the data model is not specifically tailored towards service specification, but rather centered around the artefacts in the repository. There is no fixed hierarchy of classes but registry object can be associated with each other using association classes with have an association type attribute. The attribute can have any of the predefined values or a custom value.

The ebXML standard requires implementations to support data validation for example of XML data against an XML schema.

Participant identifiers can be recorded using a ExternalIdentifier object that refers to a classification scheme. Participant contact information can be stored using pre-defined classes.

The participant type can be stored as a classification of the service instance. Services can be associated with one or more service bindings that contain zero or more references to documents that specify the technical footprint.

Functionality
The standard support advanced querying features and obviously allows the insertion and update of data in the repository. Querying may be supported through SQL syntax or a syntax specified in the ebXML standard and referred to as Filter Query syntax. It is also possible to access resources directly of HTTP using a specific URL. Queries can be stored in the registry for later use.

It also comes with notification support. Users can subscribe to events and be notified on the occurrence of the events. The subscription system uses stored queries for event selection. Notification be both pushed to, and pulled in by the user. By default the registry supports push delivery by mail and by HTTP request.

Mechanisms for data replication and federaction are also included in the standard. Furthermore the registry has build-in support for audit trails and content lifecycle management and version control.

Access control
The specification includes advanced access control policy. Policies can be specified using XACML, an open XML based standard for specifying authorization policies. The policies map users, actions and content to authorization decisions.

Interface
The interface is XML-based and both a REST-style as well as a SOAP-style interface are provided. SOAP message security is supported, both signatures and encryption.

ebXML requires certificates to be included in the message, while SIX does not allow this and instead requires that only the thumbprint of the certificate is included. Also, ebXML RR does not allow SOAP security in combination with SSL, while SOAP requires this combination.

15 Appendix D: SIX:2503 Directory protocol standard (main deliverable)

This document defines the SIX:2503 Directory protocol as part of the SIX Standards.

15.1 Status

This document is final draft. Version: 1.0.0

15.2 Introduction

15.2.1 SIX Standard

This document is part of the SIX Standards document suite.

15.2.2 Terminology

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in IETF RFC 2119 [RFC2119].

Terminology used throughout this specification conforms to that of [ebRIM] and [ebRS]

15.2.3 Notational conventions

Non-normative exemplary XML notation of the data model is provided. This XML notation is in accordance with ebXML Schema definitions [RR-CMS-XSD], [RR-LCM-XSD], [RR-RIM-XSD], [RR-RS-XSD], [RR-QM-XSD]

15.2.4 General

This specification is based on the [ebRIM] en [ebRS] specifications.,

REQ1 SIX:2503 Directory protocol implementations MUST adhere to [ebRS] en [ebRIM] specifications.

REQ2 SIX:2503 Directory protocol implementations MUST support the 'SIX:2503' Conformance profile as defined as part of this specification. (see 15.5)

15.3 Configuration

The directory lifecycle consists of two consecutive phases: the configuration phase and the operation phase. This section contains the specification of the initial configuration of the ebXML registry and repository prior to operational use.

Configuration of the registry consists of the submission of data objects that must be present in the registry prior to the start of operational use and other configuration of the ebXML registry deployment.

The data objects to be submitted include:

- Users
- Identification schemes
- Classification schemes
- Custom access control policy sets
- Parameterized stored queries

15.3.1 Pre-configured users

The following users (accounts) must be set-up during configuration.

The participant managing user is responsible for assigning participant identifiers to Organizations and participant role classifiers to Services. The participant manager also does the initial submission of new Organizations to the registry.

REQ3 A participant managing user MUST be created.

The service specification managing user is responsible for maintaining the collection of service interface specifications as determined by the Scheme organization.

REQ4 A service specification managing user MUST be created.

15.3.2 Participant identification and classification schemes

A classification scheme for participant identification schemes and participant role classification schemes is needed to designate objects as representing a participant identifier or role identifier.

```
<ClassificationScheme
          id="SIXParticipantIdentificationScheme-id"
          name="SIXParticipantIdentificationScheme" (...) >
(...)
</ClassificationScheme>
```

Code fragment 1: SIX Participant Identification Scheme

REQ5 A Classification Scheme conform Code fragment 1: SIX Participant Identification Scheme MUST be submitted to the registry by the Participant managing user.

Below this classification schemes the following classification nodes:

Participant role classification scheme

This classification node classifies a classification scheme as being a scheme that defines the participant roles in the context of the SIX directory protocol. For example: within the network (scheme) of iDEAL two participant roles are identified: issuer and acquirer. Services registered in the directory for this network must be classified as being either an issuer service or an acquirer service. To achieve this classification a classification scheme named for example 'iDEAL participant role classification scheme' is defined in the registry. This scheme is in turn classified as being a classification scheme defining participant roles in the context of the SIX directory protocol by associating the SIX Participant Role classification node with this classification scheme.

```
<ClassificationNode
          id="SIXParticipantRole-id"
          code="SIXParticipantRole"
          parent="SIXParticipantIdentificationScheme-id" (...) />
```

Code fragment 2 SIX Participant Role classification node

REQ6 A Classification Node conform Code fragment 2 MUST be submitted to the registry by the Participant Managing user.

Participant identification scheme

Classification node: SIX participant identifier. This classification node classifies a classification scheme as being a scheme that defines the participant identifier in the context of the SIX directory protocol. For example: within the network (scheme) of iDEAL each bank is assigned a specific identifier. Organizations registered in the directory for this network must be assigned this identifier. To achieve this assignment a classification scheme named 'iDEAL participant identifier classification scheme' is defined in the registry. This scheme is in turn is classified as being a Identification scheme defining participant idenitied in the context of SIX by associating the SIX participant identifier node with this specification scheme.

```
<ClassificationNode
          id="SIXParticipantIdentifier-id"
          code="SIXParticipantIdentifier"
          parent="SIXParticipantIdentificationScheme-id" (...) />
```

Code fragment 3 SIX Participant Identifier classification node

REQ7 A Classification Node conform Code fragment 3 SIX Participant Identifier classification node MUST be submitted to the registry by the Participant Managing user.

15.3.3 SIX information objects classification scheme

This specification defines a number of information objects. A classification scheme for information items defines the classification nodes that represents the different types of information objects.

```
<ClassificationScheme
          id="SIXInformationObjectsScheme-id"
          name="SIXInformationObjectsScheme" (...) >
(...)
</ClassificationScheme>
```

Code fragment 4 SIX Information Objects classification scheme

REQ8 A Classification Scheme conform Code fragment 4 SIX Information Objects classification scheme MUST be submitted to the registry by the Service specification managing user.

Below this classification schemes the following classification nodes must be specified:

Authentication information classification node

This classification node is used to classify specification links. It indicates that the classified Specification Link refers to authentication information in the context of the SIX directory protocol.

```
<ClassificationNode
     id="SIXAuthenticationInformation-id"
     code="SIXPAutheniticationInformation"
     parent="SIXInformationItemsScheme-id" (...) />
```

Code fragment 5: SIX Authentication information classification node

REQ9 A Classification Node conform Code fragment 5: SIX Authentication information classification node MUST be submitted to the registry by the Service specification managing user.

Addressing information classification node

This classification node is used to classify specification links. It indicates that the classified specification link refers to addressing information in the context of the SIX directory protocol.

```
<ClassificationNode
    id="SIXAddressingInformation-id"
    code="SIXPAddressingInformation"
    parent="SIXInformationItemsScheme-id" (...) />
```

Code fragment 6 SIX Addressing Information classification node

REQ10 A Classification Node conform Code fragment 6 SIX Addressing Information classification node MUST be submitted to the registry by the Service specification managing user.

Availability information classification node

This classification node is used to classify specification links. It indicates that the classified specification link refers to availability information in the context of the SIX directory protocol.

```
<ClassificationNode
    id="SIXAvailablityInformation-id"
    code="SIXPAvailabilityInformation"
    parent="SIXInformationItemsScheme-id" (...) />
```

Code fragment 7 SIX Availability Information classification node

REQ11 A Classification Node conform Code fragment 7 SIX Availability Information classification node MUST be submitted to the registry by the Service specification mamaging user.

15.3.4 Service status indicator classification scheme

This classification scheme defines the service availability indicator values in the context of the SIX directory protocol.

```
<ClassificationScheme
    id="SIXServiceState-id"
    code="SIXServiceState" (...) />
       (...)
</ClassificationScheme>
```

Code fragment 8 SIX Service state classification scheme

REQ12 A Classification Scheme conform Code fragment 8 SIX Service state classification scheme MUST be submitted to the registry by the Service specification managing user.

Service status indicator value classification node

This classification node indicates the current status of a service. Two values are identified:

- Available
- Unavailable

```
<ClassificationNode
        id="Available-id"
        code="Available"
        parent="SIXServiceState-id" (...) />

<ClassificationNode
        id="Unavailable-id"
        code="Unavailable"
        parent="SIXServiceState-id" (...) />
```

Code fragment 9 SIX Service state indicator classification nodes: available (a) and unavailable (b)

REQ13 A Classification Node conform Code fragment 9 SIX Service state indicator classification nodes (a) MUST be submitted to the registry by the Service specification managing user.

REQ14 A Classification Node conform Code fragment 9 SIX Service state indicator classification nodes (b) MUST be submitted to the registry by the Service specification managing user.

15.3.5 Scheme-specific configuration

For each scheme specific classification schemes for participant identification and classification SHOULD be set up.

The scheme-specific participant identification scheme SHOULD itself be classified as a SIX Participant identification scheme using the Classification node specified in REQ7.

The scheme-specific participant role classification scheme SHOULD itself be calssified as a SIX Participant role classification scheme using the Classification node specified in REQ6.

For example (for the iDEAL scheme):

```
<ClassificationScheme  id="iDEALIdentfierScheme-id" name="iDEALIdentifierScheme" (...)>

          <!-- Reference to the generic participant identification scheme -->
          <Classification
                classificationScheme="SIXParticipantIdentification"
                classifiedObject="iDEALIdentifierScheme-id"
                classificationNode="SIXParticipantIdentifier" (...) />

</ClassificationScheme>

<ClassificationScheme id="iDEALRolesScheme-id" name="iDEALRolesScheme"  (...)>

    <!-- Reference to the generic participant role classification scheme -->
          <Classification
                classificationScheme="SIXParticipantIdentification"
          classifiedObject="iDEALRolesScheme-id"
                classificationNode="SIXParticipantRole" (...) />

    <!-- Defining the different roles in this particular scheme -->
          <ClassificationNode parent="iDEALRolesScheme-id" name="iDEALIssuer" (...) />
          <ClassificationNode parent="iDEALRolesScheme-id" name="iDEALAcquirer" (...)
/>

</ClassificationScheme>
```

Code fragment 10 Example scheme-specific configuration (non-normative)

15.3.6 Stored queries

For convenience the stored query mechanism provided by EBXML registries can be leverage for easy querying. This mechanism allows predefinition and parameterization of queries.

The following parameterized stored queries SHOULD be submitted to the registry.

Query to retrieve all information on a single participant

This query retrieves all information related to one participant with a specific participant identifier. In EBXML registry terms this retrieves all objects owned by the Organization that is identified by the external identifier with a specific value and from the scheme-specific identification scheme.

```
<!-- Find all objects... -->
<RegistryObjectQuery>

    <!-- ...that are... -->
        <SourceAssociationQuery>

            <!-- ...owned by... -->
        <AssociationTypeQuery>
                <PrimaryFilter
                    comparator="LIKE"
                    domainAttribute="id"
                    value="ResponsibleFor-id"
                    xsi:type="StringFilterType" />
            </AssociationTypeQuery>

            <!-- ...the organisation with $participantId -->
            <RegisteryObjectQuery>
                <ExternalIdentifierQuery>
                    <PrimaryFilter
                        comparator="LIKE"
                        domainAttribute="value"
                        value="$participantId"
                        xsi:type="StringFilterType" />
                    <IdentificationSchemeQuery>
                        <ClassficiationQuery>
                            <ChildrenQuery>
                                <PrimaryFilter
                                    comparator="LIKE"
                                    domainAttribute="id"
                                value="SIXParticipantIdentifier-id"
                                    xsi:type="StringFilterType" />
                                </ChildrenQuery>
                        </ClassficiationQuery>
                    </IdentificationSchemeQuery>
                </ExternalIdentifierQuery>
            </RegisteryObjectQuery>

        </SourceAssociationQuery>

</RegistryObjectQuery>
```

Code fragment 11 Single participant stored query definition

REQ15 A query conform Code fragment 11 Single participant stored query definition SHOULD be submitted to the registry.

Query to retrieve all information on all participants in a specific role

This query retrieves all information related to all participants in a specific role. IN EBXML terms: all objects owned by Organizations that are classified by classification with $participantRoleId within the scheme-specific participant role classification scheme.

```
<!-- Find all objects... -->
<RegistryObjectQuery>

          <!-- ...that are... -->
          <SourceAssociationQuery>

                <!-- ...owned by... -->
                <AssociationTypeQuery>
                      <PrimaryFilter
                            comparator="LIKE"
                            domainAttribute="id"
                            value="ResponsibleFor-id"
                            xsi:type="StringFilterType" />
                </AssociationTypeQuery>

                <!-- ...organizations… -->
                <RegisteredObjectQuery>

                <!-- ...classified with $participantRoleId -->
                      <ClassificationQuery>
                            <PrimaryFilter
                                  comparator="LIKE"
                                  domainAttribute="value"
                                  value="$participantRoleId"
                                  xsi:type="StringFilterType" />

                            <ClassificationSchemeQuery>
                                  <ClassficiationQuery>
                                        <ChildrenQuery>
                                              <PrimaryFilter
                                               comparator="LIKE"
                                               domainAttribute="id"
                                               value="SIXParticipantRole-id"
                                               xsi:type="StringFilterType" />
                                        </ChildrenQuery>
                                  </ClassficiationQuery>
                            </ClassificationSchemeQuery>

                </ClassificationQuery>

                </RegisteredObjectQuery>

          </SourceAssociationQuery>
</RegistryObjectQuery>
```

Code fragment 12 Multiple participants stored query definition

REQ16 A query conform Code fragment 12 Multiple participants stored query definition SHOULD be submitted to the registry.

Query to find events for real-time notifications

The real-time notification mechanism used requires a query the yields all relevant events.

REQ17 A query conform Code fragment 13 Notification events stored query definition MUST be submitted to the registry.

Select events that require a real-time notification

```
<AuditableEventQuery>
     <AffectedObjectQuery>
                <ClassificationSchemeQuery>
                      <PrimaryFilter
                      comparator="LIKE"
                      domainAttribute="name"
                            value="SixServiceStateScheme"
                            xsi:type="StringFilterType" />
                </ClassificationSchemeQuery>
           </AffectedObjectQuery>
</AuditableEventQuery>
```

Code fragment 13 Notification events stored query definition

15.3.7 Access control policies

REQ18 The directory MUST support custom access control policies based upon the normative binding of the ebXML RS Access Control Model to [XACML].

REQ19 The directory MUST use the custom access control policies defined below as default access control policy. (The directory MUST NOT use the default access control policy as specified in [ebRIM])

REQ20 The following list summarizes the Access Control Policy semantic that a registry MUST implement:

- An unauthenticated client is denied all access
- Only a registered user is granted access to read actions on all items.
- Only a registered user is granted access to all actions on objects it is responsible for.
- Only the participant managing user can classify an Organization in the classification scheme SixParticipantRole or SixParticipantIdentifier
- Participants may reference information object classification nodes submitted by the service specification provider.
- Participants may reference service specifications submitted by the service specification provider.

Subjects / users

The following subjects are identified:

- Participants
- Service specification provider
- Participant manager

Rules

The following rules are identified:

- All everything to Registry administrator user (default access policy)
- Permit all actions on owned objects to owners

- Permit read access on all objects to all registered users, deny to guest user
- Permit reference action to all objects, except objects owned by the participant managing user

REQ21 The XACML policy conform Code fragment 14 XACML policy set specification MUST be submitted to the registry.

```
<?xml version="1.0" encoding="UTF-8"?>

<PolicySet
     PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policycombining-
           algorithm:permit-overrides"
     PolicySetId="urn:oasis:names:tc:ebXMLregrep:3.0:rim:acp:policy:folderACP1"
     xmlns="urn:oasis:names:tc:xacml:1.0:policy"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="urn:oasis:names:tc:xacml:1.0:policy
           cs-xacmlschema-policy-01.xsd">

     <Description>Default PolicySet for SIX:2503</Description>
     <Target>
           <Subjects>
                 <AnySubject/>
           </Subjects>
           <Resources>
                 <AnyResource/>
           </Resources>
           <Actions>
                 <AnyAction/>
           </Actions>
     </Target>

     <!-- Allow everything to Registry administrator user -->
     <PolicyIdReference>
           urn:oasis:names:tc:ebXMLregrep:3.0:rim:acp:policy:policyid:
           permit-registryadministrator-all
     </PolicyIdReference>

     <!-- Permit all actions on owned objects to owners -->
     <PolicyIdReference>
           urn:oasis:names:tc:ebXMLregrep:3.0:rim:acp:policy:policyid:
           permit-owner-all
     </PolicyIdReference>

     <!-- Permit read action on to all registered user, except guest -->
     <Policy
           PolicyId="urn:oasis:names:tc:ebXMLregrep:3.0:rim:acp:policy:
                 policyid:permit-registeredusers-to-read"
           RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combiningalgorithm:
                 permit-overrides">
           <Target>
                 <Subjects>
                       <AnySubject/>
                 </Subjects>
                 <Resources>
                       <AnyResource/>
                 </Resources>
                 <Actions>
                       <AnyAction/>
                 </Actions>
           </Target>
           <Rule Effect="Permit"
                 RuleId="urn:oasis:names:tc:ebXMLregrep:3.0:rim:acp:rule:ruleid:
                       permit-anyone-to-read">
                 <Description>
                       Any Subject can perform read action on any resource.
                 </Description>
                 <Target>
                       <Subjects>
```

```
<SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:not">
    <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                urn:oasis:names:tc:ebXML-regrep:SubjectRole:RegistryGuest
          </AttributeValue>
          <SubjectAttributeDesignator
                AttributeId="urn:oasis:names:tc:ebXML-regrep:3.0:rim:acp:
                      subject:roles"
                DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
    </SubjectMatch>
</SubjectMatch>

                      </Subjects>
                      <Resources>
                            <AnyResource/>
                      </Resources>
                      <Actions>
                            <Action>

<ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
          Read
    </AttributeValue>
    <ActionAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
</ActionMatch>

                            </Action>
                      </Actions>
                </Target>
          </Rule>
    </Policy>

    <!-- Permit reference action to all objects... -->
    <Policy
          PolicyId="urn:oasis:names:tc:ebXMLregrep:3.0:rim:acp:policy:
                add-classification"
          RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:
                rule-combiningalgorithm:permit-overrides"
          xmlns="urn:oasis:names:tc:xacml:1.0:policy"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="urn:oasis:names:tc:xacml:1.0:policy
                cs-xacmlschema-policy-01.xsd">

          <Target>
                <Subjects>
                      <AnySubject/>
                </Subjects>
                <Resources>
                      <AnyResource/>
                </Resources>
                <Actions>
                      <AnyAction/>
                </Actions>
          </Target>
          <Rule Effect="Permit"
                RuleId="urn:oasis:names:tc:ebXMLregrep:3.0:rim:acp:rule:ruleid:
                      permit-all-reference-servicespecifications-rule">
                <Description>
                      Allow all participants to reference all Objects
                </Description>
```

```
                <Target>
                      <Subjects>
                            <AnySubject/>
                      </Subjects>
                      <Resources>
                            <AnyResource/>
                      </Resources>
                      <Actions>

<Action>
     <!-- Match "reference" action -->
     <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
           <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                Reference
           </AttributeValue>
           <ActionAttributeDesignator
                AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
     </ActionMatch>
</Action>

                      </Actions>
                </Target>
           </Rule>

           <!-- except objects owned by the participant managing user -->
           <Rule Effect="Deny"
                RuleId="urn:oasis:names:tc:ebXMLregrep:3.0:rim:acp:rule:ruleid:
                      deny-reference-identifiers-classifications-rule">
                <Description>
                      Deny participants to assign their own identifier and
                      participant types
                </Description>
                <Target>
                      <Subjects>
                            <AnySubject/>
                      </Subjects>
                      <Resources>
                            <ResourceMatch>
                      </Resources>
                      <Actions>

<Action>
     <!-- Match "reference" action -->
     <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
           <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                Reference
           </AttributeValue>
           <ActionAttributeDesignator
                AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
     </ActionMatch>
</Action>

                      </Actions>
                </Target>

<Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
     <AttributeValue
           DataType="http://www.w3.org/2001/XMLSchema#string">${ParticipantManagerId}
     </AttributeValue>
     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:anyURI-one-and-only">
```

```
            <ResourceAttributeDesignator
                  AttributeId="urn:oasis:names:tc:ebXMLregrep:3.0:rim:acp:resource:owner"
                  DataType="http://www.w3.org/2001/XMLSchema#anyURI" />
      </Apply>
</Condition>

            </Rule>
      </Policy>
</PolicySet>
```

Code fragment 14 XACML policy set specification

15.3.8 Authentication mechanism configuration

- Registry MUST be configured to function as the Authentication Authority
- Registry MUST use the public keys submitted by participant to authenticate the users representing the participants
- Contrary to the EBXML specification the registry MUST support a combination of HTTP/S and SOAP message security[WSS-SMS]

15.4 Operation

After the initial configuration of the registry it is put in operational use. This section contains the relevant specification for this phase.

15.4.1 Data model

The section normatively specifies the method of storing information in the directory.

Participant related data

The central entities that represent a participant and related data are the Service object, as defined in [ebRIM] with an associated ServiceBinding object and the Organisation object.

REQ22 For each participant at least one Service object and one ServiceBinding object MUST be present in the directory.

REQ23 For each participant exactly one Organisation object MUST be present in the directory.

REQ24 For each participant exactly one User object representing the participant MUST be present in the directory.

Association objects MUST be present associating the Organisation as responsible entity for the Service and ServiceBinding relating to the participant.

```
<Service id="$serviceId" (...)>
          <ServiceBinding id="$serviceBindingId" service="$serviceId">
          (...)
          </ServiceBinding>
</Service>

<Association sourceObject='$organisationId' targetObject='$serviceId' (...)
associationType="urn:oasis:names:tc:ebXML-regrep:AssociationType:ResponsibleFor" />

<Organisation id="$organisationId" (...) />

<Association sourceObject='$organisationId' targetObject='$userId' (...)
associationType="urn:oasis:names:tc:ebXML-regrep:AssociationType:AffiliatedWith" />

<User id="$organisationId" (...) />
```

Code fragment 15 Service specification

Identification data

REQ25 Phone number(s) to contact the participant MUST be stored as PhoneNumber objects below the Organisation object.

REQ26 At least one PhoneNumber object MUST be present.

REQ27 E-mail addresses to contact the participant MUST be stored as EmailAddress objects below the Organisation object.

REQ28 At least one EmailAddress object MUST be present.

Participant identification

REQ29 The scheme-specific participant identifier MUST be stored as an ExternalIdentifier object assigning the scheme-specific participant identier to the Organisation object.

REQ30 The ExternalIdentifier object used to store the scheme-specific participant identifier MUST contain the identifier in its value attribute.

REQ31 The ExternalIdentifier object registryObject attribute MUST reference the Organisation object.

REQ32 The External Identifier object indetificationScheme attribute MUST reference the ClassificationScheme object representing the scheme-specific identification scheme. (see 15.3.5)

```
<Organisation id="$organisationId" externalIdentifier="$externalIdentifierId">

     <ExternalIdentifier
           id="$externalIdentifierId"
           registryObject="$organisationId"
           identificationScheme="$identificationSchemeId"
           value="$participantId" />

</Organisation>
```

Participant role

REQ33 The participant role MUST be specified as a classification of the Organisation object.

REQ34 The Classification MUST reference the ClassificationNode representing the scheme-specific role denotation. (See 15.3.5)

REQ35 The Classification MUST reference the ClassificationScheme respresenting the scheme-specific role denotation scheme. (See 15.3.5)

REQ36 The Classification object MUST reference the Organisation object as the classified object.

```
<Organisation id="$organisationId">

    <Classification
          classificationScheme='$classificationSchemeId'
          classifiedObject="$organisationId"
          classificationNode='$classificationNodeId3' />

</Organisation>
```

Addressing data

REQ37 Addressing information MUST be stored according to the following specification.

REQ38 The participant addressing information MUST be contained in a WSDL document

REQ39 The WSDL document containing the participant addressing information MUST be submitted to the repository. The related ExtrinsicObject object must be referenced by a SpecificationLink object.

REQ40 This SpecificationLink object MUST reference the ServiceBinding representing this participants services.

REQ41 This SpecificationLink object MUST specify the mimeType-attribute as “application/xml”.

REQ42 The SpecificationLink object MUST be classified as being the specification link to the addressing information. This Classification MUST reference the canonical ClassificationNode that represents the address information designation.

```
<Service id="$serviceId">
     <ServiceBinding id="$serviceBindingId" service="$serviceId">

          <SpecificationLink
               id="$specificationLinkId1"
               serviceBinding="$serviceBindingId"
               specificationObject="$extrinsicObjectId1">
                    <Classification
                         classificationScheme='$classificationSchemeId'
                         classifiedObject='$specificationLinkId1'
                         classificationNode='$classificationNodeId1' />
          </SpecificationLink>

     </ServiceBinding>
</Service>

<ExtrinsicObject id='$extrinsicObjectId1' mimeType='application/xml' />
```

Authentication information

REQ43 Authentication information MUST consist of certificates. Certificates MUST be formatted according to the [XMLDSIG] specification.

REQ44 The [XMLDSIG] documents containing the participant authentication information MUST be submitted to the repository. The related ExtrinsicObject object must be referenced by a SpecificationLink object.

REQ45 This SpecificationLink object MUST reference the ServiceBinding representing this participant's services.

REQ46 This SpecificationLink object MUST specify the mimeType-attribute as "application/xml".

REQ47 The SpecificationLink object MUST be classified as being the specification link to the authentication information. This Classification MUST reference the canonical ClassificationNode that represents the authentication information designation.

```
<Service id="$serviceId">
     <ServiceBinding id="$serviceBindingId" service="$serviceId">

          <SpecificationLink
               id="$specificationLinkId1"
               serviceBinding="$serviceBindingId"
               specificationObject="$extrinsicObjectId1">
                    <Classification
                         classificationScheme='$classificationSchemeId'
                         classifiedObject='$specificationLinkId1'
                         classificationNode='$classificationNodeId1' />
          </SpecificationLink>

     </ServiceBinding>
</Service>

<ExtrinsicObject id='$extrinsicObjectId1' mimeType='application/xml' />
```

Availability information

Availability information consist of a document formatted according to the specification.

REQ48 The availability documents containing the participant's availability information MUST be submitted to the repository. The related ExtrinsicObject object must be referenced by a SpecificationLink object.

REQ49 The documents containing the participant's availability information MUST be a valid SIX availability information document.

REQ50 This SpecificationLink object MUST reference the ServiceBinding representing this participant's services.

REQ51 This SpecificationLink object MUST specify the mimeType-attribute as "application/xml".

REQ52 The SpecificationLink object MUST be classified as being the specification link to the availability information. This Classification MUST reference the canonical ClassificationNode that represents the availability information designation.

```
<Service id="$serviceId">
     <ServiceBinding id="$serviceBindingId" service="$serviceId">

          <SpecificationLink
               id="$specificationLinkId1"
               serviceBinding="$serviceBindingId"
               specificationObject="$extrinsicObjectId1">
                    <Classification
                         classificationScheme='$classificationSchemeId'
                         classifiedObject='$specificationLinkId1'
                         classificationNode='$classificationNodeId1' />
          </SpecificationLink>

     </ServiceBinding>
</Service>

<ExtrinsicObject id='$extrinsicObjectId1' mimeType='application/xml' />
```

REQ53 The current availability status of a service MUST be indicated in the directory.

REQ54 The current availability status MUST be indicated by adding a Classification to the Service object. This Classification references the service state ClassificationScheme and the appropriate service state value ClassificationNode. (See YYY)

```
<Service id="$serviceId">
     <ServiceBinding id="$serviceBindingId" service="$serviceId">

          <Classification
               classificationScheme='$classificationSchemeId3'
               classifiedObject="$serviceId"
               classificationNode='$classificationNodeId6' />

     </ServiceBinding>
</Service>
```

Service specification

REQ55 The service interface specifications of the services within the network MUST be submitted by the Service specification management user.

REQ56 The service interface specifications MUST be WSDL documents.

The [WSDL] document submitted to the repository MUST have associated ExtrinsicObject objects.

REQ57 The ExtrinsicObject objects MUST be classified as being service interface specifications. This classification MUST reference the preconfigured ClassificationNode representing a service interface specification.

```
<ExtrinsicObject id='$extrinsicObjectId4' mimeType='application/xml'>
     <Classification
           classificationScheme='$classificationSchemeId1'
           classifiedObject="$extrinsicObjectId4"
           classificationNode='$classificationNodeId8/>
</ExtrinsicObject>
```

15.4.2 Real-time notifications

EBXML Registry provides a subscription mechanism the allows user to subscribe to events in the lifecycles of registry objects. This mechanisms is leveraged to implemented the required real-time notification support.

The configuration of the EBXML registry provides a stored query users can subscribe to that results in the notification on all required events.

REQ58 Participants MUST create a subscription to the stored query (see section X)

REQ59 The Participant MUST NOT specify and end time for the subscription.

REQ60 The Participant MUST specify the Service Notify action (EBXML RS 7.3.2) for the subscription and provide the end-point URI. The participant MUST specify the notificationOption attribute as 'Objects'.

REQ61 The Participant MUST implement a service end-point for notification delivery.

REQ62 The registry MUST deliver notification in XML markup.

15.5 Conformance profile

Feature	Registry Lite	Registry SIX:2503	Registry Full
SOAP Binding			
QueryManager binding	MUST	**MUST**	MUST
LifeCycleManager binding	MUST	**MUST**	MUST
HTTP Binding			
RPC Encoded URL	MUST	MUST	MUST
User Defined URL	MAY	MAY	MUST
File Path URL	MAY	MAY	MUST
LifeCycleManager			MUST
SubmitObjects Protocol	MUST	**MUST**	MUST
UpdateObjects Protocol	MUST	**MUST**	MUST
ApproveObjects Protocol	MUST	MUST	MUST
DeprecateObjects Protocol	MUST	**MUST**	MUST
UnderprecateObjects Protocol	MUST	**MUST**	MUST
RemoveObjects Protocol	MUST	**MUST**	MUST
Registry Managed Version Control	MAY	**MUST**	MUST
QueryManager			
SQL Query	MAY	**MUST**	MUST
Filter Query	MUST	**MUST**	MUST
Stored Parameterized Query	MAY	**MUST**	MUST
Iterative Query	MUST	**MUST**	MUST
Event Notification	MAY	**MUST**	MUST
Content Management Services			
Validate Content Protocol	MAY	MAY	MUST
Catalog Content Protocol	MAY	MAY	MUST
Canonical XML Cataloging Service	MAY	MAY	MUST
Cooperating Registries			
Remote object references	MAY	MAY	MUST
Federated queries	MAY	MAY	MUST
Object Replication	MAY	MAY	MUST
Object Relocation	MAY	MAY	MUST
Registry Security			
Identity Management	MUST	**MUST**	MUST
Message Security			
- Transport layer security	MAY	**MUST**	MUST
- SOAP Message Security	MUST	**MUST**	MUST
Repository Item Security	MUST	**MUST**	MUST
Authorization and Access Control			
- Default Access Control Policy	MUST	**MUST**	MUST
- Custom Access Control Policies	MAY	**MUST**	MUST
Audit Trail	MUST	**MUST**	MUST
Registry SAML Profile	MAY	MAY	MUST
NLS	MUST	**MUST**	MUST

15.6 Normative references

[RFC2119] S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, IETF RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt.

[ebRIM] ebXML Registry Information Model Version 3.0 http://www.oasis-open.org/committees/regrep/documents/3.0/specs/regrep-rim-3.0-cs-01.pdf

[ebRS] ebXML Registry Services Specification Version 3.0 http://www.oasis-open.org/committees/regrep/documents/3.0/specs/regrep-rs-3.0-cs-01.pdf

[RR-CMS-XSD] ebXML Registry Content Management Services XML Schema http://www.oasis-open.org/committees/regrep/documents/3.0/schema/rim.xsd

[RR-LCM-XSD] ebXML Registry LifeCycleManager XML Schema http://www.oasis-open.org/committees/regrep/documents/3.0/schema/lcm.xsd

[RR-RIM-XSD] ebXML Registry Information Model XML Schema http://www.oasis-open.org/committees/regrep/documents/3.0/schema/rim.xsd

[RR-RS-XSD] ebXML Registry Service Protocol XML Schema http://www.oasis-open.org/committees/regrep/documents/3.0/schema/rs.xsd

[RR-QM-XSD] ebXML Registry QueryManager XML Schema http://www.oasisopen.org/committees/regrep/documents/3.0/schema/query.xsd

[SOAP11] W3C Note. Simple Object Access Protocol, May 2000 http://www.w3.org/TR/SOAP

[WSDL] W3C Note. Web Services Description Language (WSDL) 1.1 http://www.w3.org/TR/wsdl

[XML] T. Bray, et al. Extensible Markup Language (XML) 1.0 (Second Edition). World Wide Web Consortium, October 2000. http://www.w3.org/TR/REC-xml

[XMLDSIG] XML-Signature Syntax and Processing http://www.w3.org/TR/2001/PR-xmldsig-core-20010820/

[WSI-BSP] WS-I: Basic Security Profile 1.0 http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2004-05-12.html

[WSS-SMS] Web Services Security: SOAP Message Security 1.0 http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-messagesecurity-1.0.pdf

[WSS-SWA] Web Services Security: SOAP Message with Attachments (SwA) Profile 1.0 http://www.oasis-open.org/apps/org/workgroup/wss/download.php/10902/wssswa-profile-1.0-cd-01.pdf

16 Appendix E: About Innopay and the SIX foundation

For their internet site: "Innopay is an independent full service consultancy firm specialized in electronic payments and related financial services. Key focus areas include online payment, mobile payment, e-invoicing and e-identity. Our practice covers strategy & business development, product development & management and knowledge transfer. We use a multi-disciplinary approach covering the commercial, operational and technical aspects. Combined with our strong and proven project management capabilities we have successfully taken payment products and services for our clients from 'powerpoint to production'" (45).

The experienced consultants of Innopay have an impressive track-record in the field of 4-party network schemes. They have been involved in the creation of the online payments scheme: iDEAL, the online billing scheme: Standaard Digitale Nota and are currently involved in the revision of the e-identity scheme for businesses of the Dutch government: Digid.

Their involvement in these projects allows first hand access to the knowledge of experts in the field as well as relevant experience in developing schemes. Also access to extensive documentation regarding the existing schemes and the ability to use and adapt these to validate results is provided.

While Innopay is currently the IP holder for the SIX standard it plans to establish a non-profit SIX foundation and transfer all SIX related IPR to this foundation. The foundation will independently govern and maintain the SIX standards. Innopay is currently in the process of seeking 5 initial co-founders for this foundation.