Trust broker Framework

The Bachelor Series

Jericho in depth…

Trust broker Framework

Enabling trust in Jericho networks

Adriaan Bruning

Capgemini's Security & Innovation Research Centre, based in the Netherlands, focuses on near future IT Security solutions. The Jericho forum's vision on network de-perimeterization and Boundaryless Information flow™ has been the starting point for this research centre. Research papers from this centre appear in two distinct categories;

The Master Series
Researcher holding a masters degree in Informatics or are in the process of obtaining a master degree publish in the Master Series. The participating University and the Capgemini Security & Innovation Research centre have approved publications in this category.
Publications in this Series for 2008

- Jericho in depth… Secure Communications by A. Stan
- Jericho in depth… The road to Jericho by A. Stan

Planned publications in this Series for 2008

- Demystifying trust by F. van Leijden
- Jericho in depth… Automated Security Classification by K. Clark
- Jericho in depth… Trust Management for Trust brokers by A. Demarteau

The Bachelor Series
Researchers holding a bachelors degree in Informatics or in the process of obtaining a bachelors degree publish in the bachelor series. Their University and the Capgemini Security & Innovation Research centre have approved publications in this category.
Publication in this series for 2008

- Jericho in depth… Endpoint security by L. Teheux
- Jericho in depth… Authentication and Accounting by E. Baranikov
- Jericho in depth… Trust broker Services by A. Bruning
- Jericho in depth… Trust broker framework by A. Bruning

Planned publications in this series for 2008

- Jericho in depth… Controlling the COA framework by J. Willemsen
- Jericho in depth… Fully ASP based by D. Hanenberg & F. Aardoom

Preface

Since the dawn of the Internet at the ending of 1969 a lot has changed, I'm sure nobody will disagree with a statement like that. During the last couple of years however, we seem to have hit a mid-life crisis of the Internet. The sudden boost of Internet technology over the past decade does not fit well with our outdated design principles for network security. Most organizations hold tight to their fortress approach in trying to protect the internal network from the hostile Internet. This is understandable, but not really realistic. In the Netherlands, we are particularly proud of our water management techniques. In a country that lays for more then sixty percent below sea level we know that we have to build and maintain solid dikes to prevent our country from flooding. Having holes in these dikes quickly diminishes the whole purpose of have a dike. The same holds true for perimeter defence in computer networks. Information leakage via email, Hyves, my space or mobile data solutions like iPod or USB diminishes the purpose of perimeter security. Today's business world is one of collaboration, one of working together., one of global markets. The Internet is the ideal candidate to support this collaboration. The Jericho Forum (Open Group), formed by Security professionals from the largest organisations in the world described their vision of network de-perimeterization and boundryless Information flow™ in various publications. These visions formed the starting point for Capgemini's Security & Innovation Research Centre.

Together with the best universities in the Netherlands, Capgemini's offers academic researchers and graduate students to ability to conduct empirical academic research into the topic of Collaboration Oriented Architectures or to conduct feasibility studies into the Jericho Forums visions.

Marco Plas

Head of Jericho Research
Capgemini Security & Innovation Research Centre
Capgemini Netherlands

Executive summary

As society is changing with ever increasing speed and communication and collaboration are the key factors to more efficiency, and therefore an increase in revenue. Companies want to adapt to this new trend by increasing the interaction and exchange of information between partners. However in doing this they are held back by a number of issues. The Jericho Forum discusses all of these issues. One of the main issues is establishing a trust relationship in a digital environment. My previous research showed that trust is a vital issue in establishing communication between different and sometimes unacquainted parties. To create a trust relationship, a trust management system is needed. At Capgemini they have taken this concept to the next level and named it 'Trust broker'. A Trust broker is a trusted entity that defines a level of trust based on four subjects, these are: identity, reputation, behaviour and control. They are needed to determine the trustworthiness of someone, are implemented in the Trust broker framework. As a supplement to "*Jericho in depth... Trust broker services*"; the Trust broker framework will be further explained by using four phases of the Integrated Architecture Framework (IAF). Throughout these phases the Trust broker framework – as suggested in Trust broker services – is made more specific and consequently more applicable in order to create a trust management system. The four phases that are dealt with are the contextual, conceptual, logical and physical phase. Whereas the contextual and physical phase are a recapitulation of Trust broker services, the conceptual and logical phase present a vision concept how a Trust broker framework could be implemented. The logical phase uses this vision concept and translates it – with the use of a certain context – to the three different Trust broker models. After this a remaining problem of the security within this framework of services is discussed. This problem is securing an orchestration of services, which is the main idea of Service Oriented Architecture, as a solution three options are given. Within the logical phase a technology matrix is given. This matrix makes some suggestions regarding which technology can be used to implement the main services as defined in the conceptual phase. Furthermore, the matrix gives a readiness level to each technology which is based on ten requirements. Finally, the Trust broker framework is made more tangible by describing three use-case scenarios that are coupled to the business scenarios as given by the Jericho Forum.

Contents

Introduction

In my previous book, *'Jericho in depth... Trust broker services'*, I have shown that society is extremely dependent on the Internet. However, the blueprint of the web is based on the ideas that originated in the sixties and seventies from last century. It was not designed to be scaled up in this manner and was only designed for one specific user: the U.S. Army. The only thing important to them was that nobody could get access to this network. Sadly – after more than thirty years - this is still how we protect our virtual assets, by denying people access to these networks where the valuable assets stored. However, due to a change in business demands it is not longer desirable – and in some cases not even profitable - to keep everything to yourself. Nowadays the key to success is collaboration, collaboration between people, companies, governments and everything in between. Nevertheless, in establishing these collaborations people – especially IT and business people – are confronted with a lot of security issues. If they do collaborate in the way they want and share their information the chances are high it gets misused, lost or stolen. All of these options are not preferable – at the very least – but since some major accounting scandals in the U.S.A., i.e. Enron, the law has made some important regulations that make these flaws in security even more dreadful. Because people accountable are now be severally liable for these flaws in security. The most important regulation today – that affects this field – is the Sarbanes-Oxley Act. So to get success companies have to collaborate, but are held back because of security and other regulations. To amend this Jericho Forum has proposed a radical new approach for security; de-perimeterization. In short, don't protect the network, but protect the valuable assets. The Jericho Forum has made this proposal in 2004 and has produced a lot of white and positioning papers, but never produced a actual prototype. This daring task has been taken up by Capgemini Netherlands b.v. Since early 2007 a team has been researching how the ideas of the Jericho Forum can be made more applicable. This team has come to the same conclusion as the Jericho Forum, in order to collaborate securely you have to be able to make or get a valued judgment of the entity you want to collaborate with. In other words, we need to develop a notion of trust in a digital environment, hence a trust management system. This trust management system is presented in my previous book as a Trust broker. During this research it became clear that the Trust broker would become the metaphorical spider in the web. By getting such a central important role the need to create a secure trustworthy system only magnified. In order to accomplish this one commandment of the Jericho Forum was used, - JFC number 10, *"security of any asset of sufficiently high value requires a segregation of duties"*. This principle has lead to the creation of the Trust broker framework.

In this book will further extend the research of the Trust broker framework and I shall endeavor to formulate a way to make this concept applicable and perhaps ready for a prototype.

Adriaan Bruning

1. Aims of research

Aims

As an extended research, this book will be a continuation of the previous book, *'Jericho in depth... Trust broker services'*, and deliver a next phase to this research. The aim is to define the specific requirements needed to create an IT architecture, or in this case a Security Architecture.

To achieve this aim the following goals will be pursued:

- Trust broker principles must be worked out according to the four phases of the Integrated Architecture Framework (IAF)
- In support of these phases a scenario will be worked out

These goals will answer to the main assignment, namely, to design the Trust broker Framework, according to a recognized framework standard, on a business and information level, given the vision and commandments by the Jericho Forum.

Research method

As described above, this research will be done with the help of the Integrated Architecture Framework (IAF). This model is chosen because of several reasons:

1 Initially the TOGAF framework would be used. But studies showed me this is a far more procedural focused framework, that is well suited to design a complete IT infrastructure but not very suitable to design a high level infrastructure that is focused on security. Therefore, I have chosen a more generic framework that is more focused on the business goals, namely IAF.

2 Besides IAF being a more generic framework, my decision was also influenced by Capgemini because their standard framework is IAF. More information about IAF is given in chapter 2 and in Appendix A – Methodology.

The main goal of this research paper is to create a solid foundation for the physical layer. This means that the central focus will be to implement the existing principles into the contextual and the conceptual phase and to create the logical phase. In order to clarify certain processes of the Trust broker framework a business scenario will be created and explained. The aim will be to create the first three phases of the IAF framework, namely contextual, conceptual and logical. The physical phase will not be dealt with in a great detail, because it will be different in every situation. It is therefore more important to deliver the specific requirements needed to know how and with which technologies such a security architecture can be built. To accomplish this I will create a technology matrix for the Trust broker framework within the physical phase.

Structure

This book is structured in three chapters and supported by four appendixes. This chapter deals with the aims, structure and repeats the most important issues of the previous book. The Integrated Architecture Framework is described in chapter two. This chapter will be divided in five sub-chapters, namely contextual, conceptual, logical, physical, physical and scenarios. Chapter three will contain the conclusion and reflection of this book.

Introduction previous book

The previous research book, 'Jericho in depth... *Trust broker Services'* will largely influence this research assignment. The results will be used to determine the next/ second phase, and if needed additional information will be created and added. In this sub-chapter I shall give an overview of the most important results of the book 'Jericho in depth... *Trust broker services'*.

Our trust in another individual can be based on our evaluation of his or her ability, integrity, and benevolence. Digital trust is based on the same principles. Yet the process of how to decide is completely different. As a result the problem, evaluating the integrity information about another entity in a digital environment is very difficult. There are numerous reasons for this. The most important reason is that one can never be certain with whom one is communicating, let alone the background of this entity. In other words one needs to know specific things about the entity one is communicating with in order to decided whether you can trust this entity or not. To establish such a trust relation you need to know four things in order to determine a trust level:

1. identity
2. reputation
3. skills
4. control

In order to fully use a de-perimeterized network, users must be able to trust different kinds of systems. This is enabled by the Trust broker. The Trust broker evaluates entities on the four issues listed above. On the basis of these evaluations it can give entities access to services that were normally too sensitive or dangerous to share across the web. However, due to the increased strain on security the Trust broker must be protected very well. So in order to comply with regulations and to prevent it from becoming a single point of failure (SPOF) the Jericho forum commandment rule number 10 is used. This rule says that to secure something of great value you must apply a segregation of duties. To apply the segregation of duties principle, the Trust broker Framework was devised. This framework has a modularly build-up that can deliver any type of service over the internet. To deliver these services the service oriented architecture (SOA) is used in combination with a trust management system, thus a Trust broker. More about SOA and the Trust broker can be found in Appendix B – SOA and the Trust broker. As a conclusion I said that the Trust broker is possible to implement, but the required technology must first be brought to a higher level of maturity in order to really build a prototype.

2. Trust broker framework

As shown in my previous research – main conclusions are described in chapter 10 – the need for a framework that deals with all the different functionalities is necessary. In this chapter I shall explain this framework by using IAF. After completing all four phases there will be a clearer vision about the details of the Trust broker Framework, its functional requirements and logical components.

Integrated Architecture Framework

As described in chapter 1, the Integrated Architecture Framework (IAF) is a generic framework that is used to structure and define the architectural content. As shown in figure 1, IAF is sub divided into four columns – the Aspect Areas – and four sectors – Abstraction levels. This partitioning creates different cells, each with their own predefined set of Artifacts.

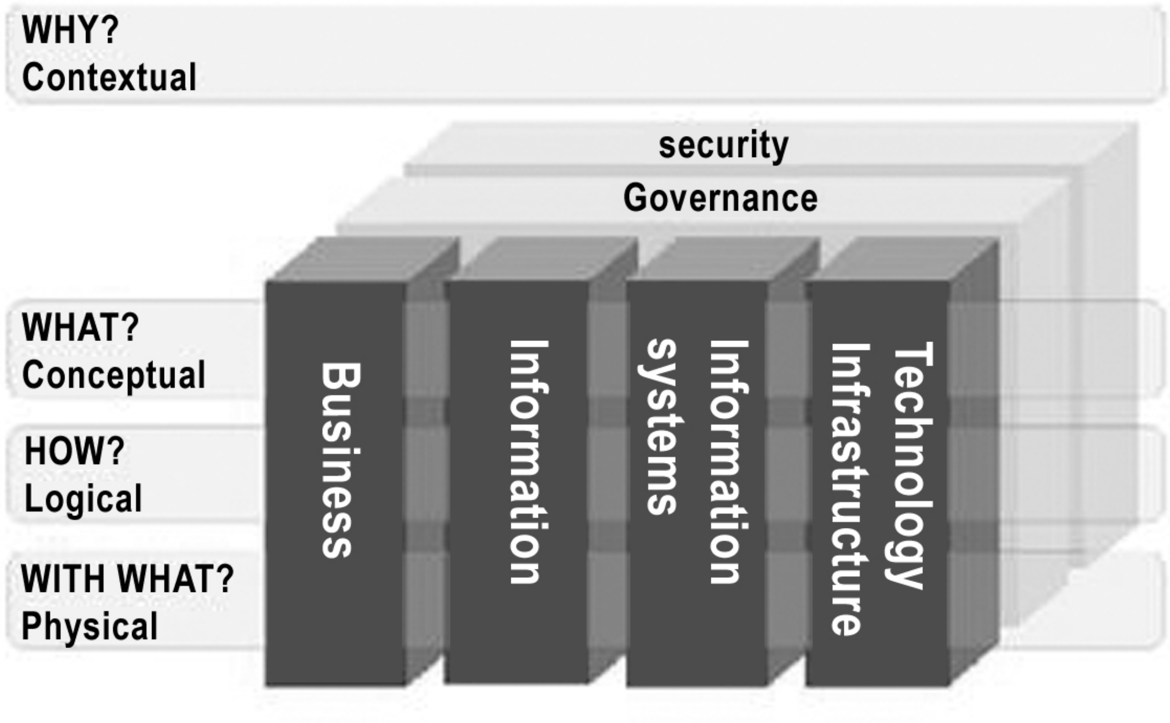


Figure 1 - The integrated Architecture Framework

I will briefly explain this model by giving an extract of a Capgemini document[1]. Capgemini uses the Integrated Architecture Framework (IAF) version 4 as a framework for their architecture engagements. IAF is used to structure and define the architecture content. The framework provides a model for architecture development and usage describes the format and content of elements of the architecture and specifies the way in which these elements relate to each other. Within this framework, IAF artifacts describe the architecture elements. Artifacts belong to, and are derived within, specific areas in the architecture framework.

[1] http://www.capgemini.com/resources/thought_leadership/architecture_and_the_integrated_architecture_framework/?d=1 *(accessed september 2007)*

Abstraction Levels

Abstraction within the IAF allows a consistent level of definition and understanding to be achieved in each area of the architecture, and is especially useful when dealing with large and complex architectures, as it allows for all relevant issues to be identified before further detailing is attempted.

The IAF defines four levels of abstraction:

- *The Contextual Level* is characterized by the "Why?" question. It is not about understanding what the new architecture will be, but identifying the boundaries for the new architecture and its context. Specifically this level focuses on the business aspirations and drivers and captures the Principles upon which the architecture will be based.
- *The Conceptual Level* is characterized by the "What?" question. The requirements and objectives are decomposed, ensuring that all aspects of the scope are explored, that relevant issues are identified and these issues are resolved without concern over how the architecture will be realized.
- *The Logical Level* is characterized by the "How?" question. The Logical Level is about finding the ideal solution in an implementation independent manner. From this, several "solution alternatives"can be developed that either provide the same outcome, or alternatively "test" different priorities and scenarios to understand the implication of different potential outcomes.
- *The Physical Level* is characterized by the "With what?" question. It is about determining the real world structure and organization, and is concerned with translating the Logical Levels 'ideal' structure and organization into an implementation-specific structure, bounded by standards, specifications and guidelines.

Since services are defined at the Conceptual Level and structured (and organized) at the Logical Level, based on outcomes (Principles) from the contextual level, the Physical Level is out of scope.

Aspect Areas

To break down the complexity of the Architecture, IAF recognizes six “Aspect Areas. ” four of which focus exclusively on the core aspects of the overall architecture; Business, Information, Information Systems and Technology Infrastructure. The remaining two aspect areas specifically address the disciplines of Security and Governance.

- The *Business Aspect Area* adds knowledge about business objectives, activities, and organizational structure. Key artifacts in this aspect area include Business Goal, Business Service, Business Actor, Logical Business Component and Physical Business Component.
- The *Information Aspect Area* adds knowledge about the information the business uses, the information structure and relationships. Key artifacts in this area include Information Object, Business Information Service, Logical Information Component, etc.
- The *Information System* Aspect Area adds knowledge about types of information systems (packaged or bespoke) that can automate and support the processing of the information used by the business. Key artifacts include IS Service, Logical IS Component and Physical IS Component.
- The *Technology Infrastructure Aspect Area* adds knowledge about types and structure of infrastructure components (“boxes and wires”) that support the information systems and actors. Key artifacts include TI Service, Logical TI Component and Physical TI component.
- The *Governance Aspect Area* adds knowledge about the manageability and quality of the architecture implementation. The artifacts for this area are all fundamentally defined within the core aspects areas.
- The *Security Aspect Area* adds knowledge about mitigating known risks to the architecture implementation. The artifacts for this aspect area are all fundamentally defined within the core aspects areas. (*Architecture and the Integrated Architecture Framework, Capgemini 2006, Andy Mulholland*)

More Information about IAF and the need for architecture can be found in Appendix A – Methodology.

Contextual

The contextual phase of the IAF framework answers the “why?” question. Its main goal is to understand the why and the overall context of the business and the architecture by answering questions like: ‘What are businesses trying to achieve, what is the scope of enterprises, what are the business drivers to attain solutions and what are the constraints?’ In the context of the Jericho Forum not all of these questions can be answered in the usual manner. As the IAF framework is intended to make enterprise architecture for one specific company. But for the Jericho Project the intention is to create a highly general architecture that all kinds of companies can use. This means that I shall interpret the required information, where necessary, to the specific needs for this book. Information needed for this phase is divided in two classes: *Supporting inputs* and *Architecture principles* The information for these two types is largely provided by the Jericho Forum, and will therefore be quoted several times.

Supporting Input

The following supporting inputs will be handled: vision, mission, strategies, business and technology context

Vision

The vision of de-perimeterization is given by the Jericho Forum is:

"To enable business confidence for collaboration and commerce beyond the constraint of the corporate, government, academic and home office perimeter, principally through:

- *Cross-organisational security processes and services,*
- *ICT products that conform to open security standards,*
- *Assurance processes that when used in one organisation can be trusted by others."*[2]

Mission

The mission statement of the Jericho Forum is:

"Act as a catalyst to accelerate the achievement of the collective vision, by:

- *Defining the problem space*
- *Communicating the collective vision*
- *Challenging constraints and creating an environment for innovation*
- *Demonstrating the market"*[3]

However this is the general mission of the Jericho Forum as an organization. The mission of the de-perimeterization trend can be formulated as the following:

"We need to develop a new security architecture and design approach that will enable businesses to grow safely and securely in an open, Internet-driven, networked world. Members of the international IT security thought-leadership group – The Jericho Forum – are working together to drive and influence development of security solutions and open standards that support de-perimeterization."[4]

Strategies

Jericho Forum has not formulated a clear strategy for implementing a de-perimeterized environment, according to their vision paper it states "*A period of three to five years for the achievement of Jericho Forum's vision, whilst accepting that its mission will continue beyond that."* However co-founder of the Jericho Forum and global information security director Paul Simmonds has formulated an implementation roadmap to de-perimeterization, which is given on the next page.

[2] www.jerichoforum.com accessed july 2007

[3] www.jerichoforum.com accessed july 2007

[4] www.opengroup.org/jericho/JF07011.pdf accessed august 2007

"The four phases to de-perimeterization:

1. *Phase one:*
   - *Now: Move outside the perimeter*
     - *Move non-corporate items outside the corporate perimeter,*
     - *Deliver external services outside the corporate perimeter,*
     - *Enable internet connected working for staff and third parties.*

2. *Phase two:*
   - *Soon: Remove hardened perimeter*
     Pervasive authenticated access, transport encryption.
     - *The border becomes a QoS boundary / cost justification*
     - *The border acts as a sieve – blocking the "lumps" only,*
     - *Parts of business connect to systems needed directly via the internet,*
     - *Connection to third parties is direct (system to system)*
3. *Phase three:*
   - *Near Future: No perimeter*
     Connection level authentication, data level encryption
     - *Move from system level authentication, to;*
       - *a Connection level authentication, and;*
       - *b Data level validation*
         - *can't connect to server / data store if no rights to the data*
         - *if you connect you can only see those files to which you have rights*
4. *Phase four:*
   - *Future: Data level Authentication*
     - *Data inherently secure,*
     - *Data will only operate in validated secure environments by authorized people."*[5]

[5] Extracted from http://www.blackhat.com/presentations/bh-europe-04/bh-eu-04-simmonds.pdf accessed aug. 2007

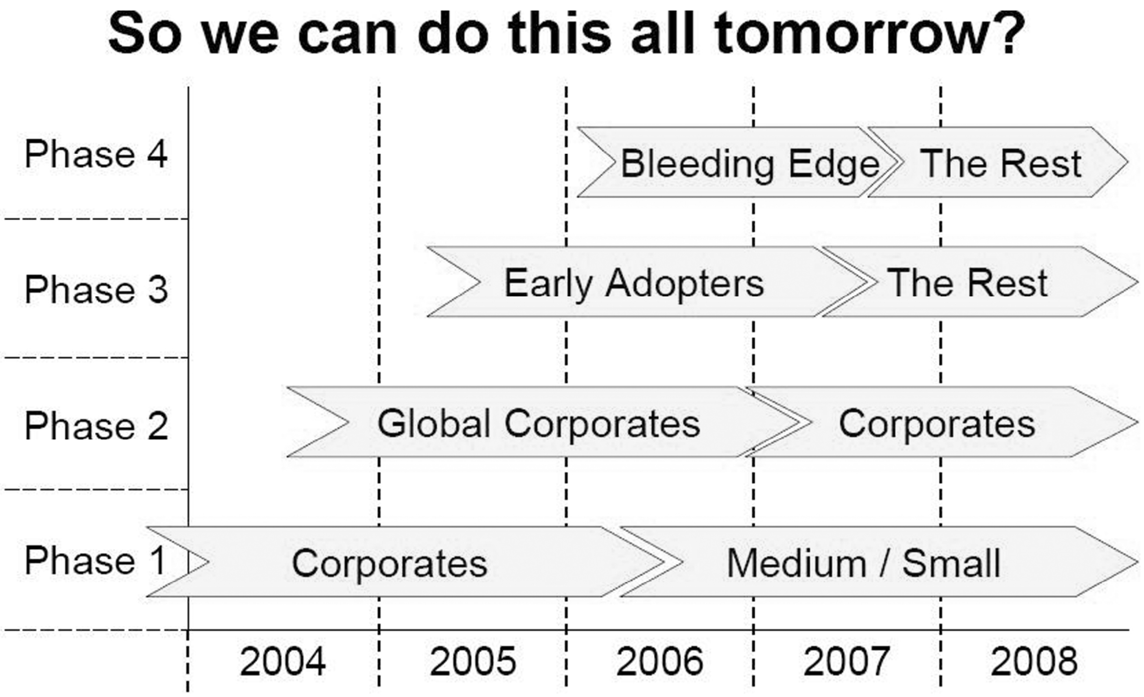


Figure 2 - Time table of de-perimeterization by Paul Simmonds:

Businesses context

In order to enhance the communication between companies and stakeholders, the standard of the necessary security attributes must be raised. Therefore, the Jericho Forum is of the opinion that the perimeter of security must be re-evaluated and probably be replaced to the objects within a company where it is really necessary, i.e. with the data. The Jericho Forum position paper about business drivers, states that this increase in collaboration is already happening. As the perimeter is detrimental to collaboration the perimeter needs to be replaced. So, in order to increase business value, the communication and collaboration between companies must become easier and safer. That is why the de-perimeterization method was created.

The business drivers for enabling a new security architecture that the Jericho Project research group, in particularly Alina Stan, have identified are:

- *"Globalization, digitalization and personalization are changing the way business is done today and forever,*
- *Wider collaboration between organizations outside their perimeters is required due to the explosion of pervasive, fast, reliable, and cheap Internet connectivity,*
- *Collaborative business environment for electronic commerce resulting in the emergence of new work patterns;*

  New business models based on electronic transactions and mobile users that connect the organizations and their business processes to all external stakeholders, sustain collaboration anytime, anywhere, at low costs among all the entities have emerged,

- *The need for trust models within the business partnerships among different organizations,*
- Business mergers and acquisitions (extended organizations) that generate new services; requirements for shorter time to market, business agility."[6]

The Jericho Forum has formulated a position paper about trust and cooperation. In this paper it give 3 motives – based on the Jericho commandments – that underpin the need for a trust management (a Trust broker) system in a de-perimeterized environment. These drivers are;

- *"Trust is crucial to all human interactions and therefore the ability to express trust* electronically is essential to successful electronic collaboration. (JFC number 6)
- Registration and Trust Management, however, are expensive and often complex due to differing policy requirements
- De-perimeterization requires the ability to share reputation information between organizations (JFC number 8)1 and thus reduce costs."[7]

In the opinion of Capgemini a trust management system in the form of a Trust broker will enable another important application. It will facilitate the collaboration between elements on the network irrelevant of their geographical location. Thus it can also become a community broker that will facilitate collaboration and communication between client and services, services and services and between users and users.

Architecture principles

Architecture context

The Jericho Forum in its vision paper gives the main architectural context of de-perimeterization. This paper is going to influence the existing architectures. The Trust broker framework will not deal with all of the parts of this vision paper. The architecture context according to the Jericho Forum:

"Constituent parts of applications/systems may be integral to that application/system, or shared,
as in the situation where enterprise architecture implements common components and ICT infrastructure used by multiple applications/systems. These parts are:

- *Process – the dynamic component of each tool or capability, or overarching business operation (process logic – the ordering and sequencing of process steps)*
- *Business Logic – the constraints and rules (including security) that must be upheld to meet business objectives; these may be embedded in applications, or implemented by other architectural components*
- *Data – both the underlying data itself, and data descriptions (meta-data) that support data communication and sharing*
- *ICT Infrastructure."*[8]

The context that the Trust broker framework is mainly focused on is the re-use of common

[6] Extracted from Alina Stan's, Jericho Security Architecture v 1.2 2007

[7] https://www.opengroup.org/jericho/trust_coop_v1.0.pdf accessed august 2007

[8] https://www.opengroup.org/jericho/vision_wp.pdf accessed july 2007

components for the enterprise architecture on a high level. The corresponding parts of the Jericho Architecture are 'Process' and 'Business Logic', which are listed above. The remaining two, 'Data' and 'ICT infrastructure', are for future detailed research, of which 'ICT infrastructure' can only be researched when the processes en business objectives are finished.

“De-perimeterisation potentially involves positioning or re-positioning security controls in any of these parts. Especially, within infrastructure there may be network security controls that de-perimeterisation repositions to host computers or other devices. The potentially affected areas are:

- *Local security components (edge controls – firewalls, routers, intrusion monitoring – and secure communications)*
- *Platforms/devices (middleware and messaging systems, database management systems, host computer operating systems, embedded operating systems)*
- *Interface standards (communications, data, security)*
- *Management frameworks (policy, identity and access, audit, incident and vulnerability)*

As discussed in section 2, the established architectural notion of ‘layering’ while providing a conceptual aid to analysis of existing security controls and design of new ones is increasingly less useful for security once considerations such as tunneling are brought into play.”[9]

The Trust broker framework has the potential to affect all lower areas, such as: 'Local security components', 'platforms/devices', 'interface standards' and 'management frameworks'. This paper will mainly concentrate on the 'management frameworks'. This choice is made because this area is focused on the more practical aspect of business. Management frameworks will show the essential operations to create a trusted – and secure – environment between entities. These essential operations will provide insight in all other areas, i.e. it will define requirements to which open standards must comply. These standards will have certain technical requirements of their own, so they will depend on a combination of specific hard and software. As a result by creating a management framework first, it will become a catalyst to all the other areas and will set in a chain reaction until all the components of an ICT infrastructure are dealt with in accordance to the Trust broker framework philosophy.

Architecture objectives

The architecture objectives are provided by the Jericho Forum in the representation of commandments and principles. In order to accomplish the vision and mission, the Jericho Forum has established a number of key principles to which any solution must comply, these are the Jericho forum commandments.

[9] https://www.opengroup.org/jericho/vision_wp.pdf accessed july 2007

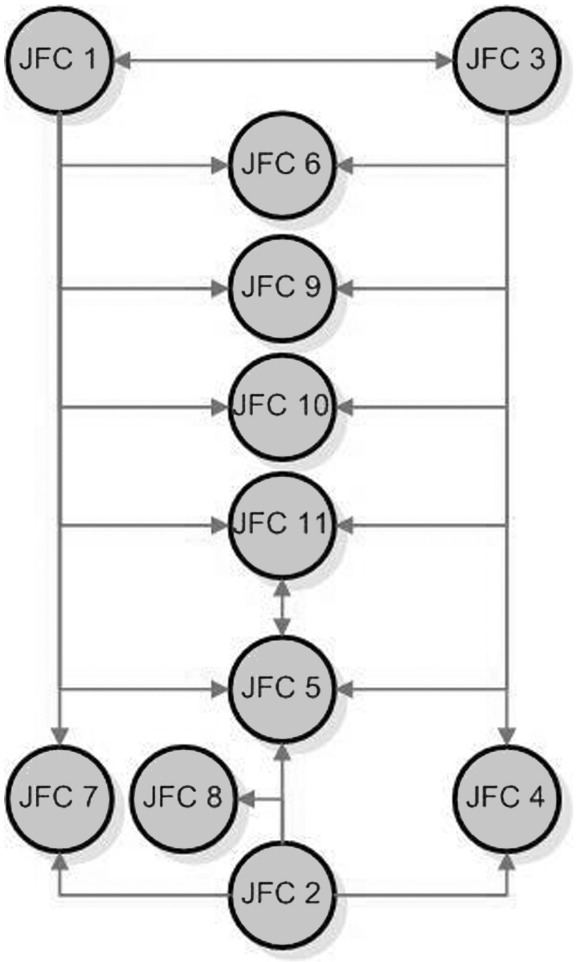


Figure 3 – The interrelationships between the Jericho Forum Commandments

Jericho Forum Commandments (JFC)[10]

1. The scope and level of protection should be specific & appropriate to the asset at risk.
2. Security mechanisms must be pervasive, simple, scalable & easy to manage.
3. Assume context at your peril.
4. Devices and applications must communicate using open, secure protocols.
5. All devices must be capable of maintaining their security policy on an un-trusted network.
6. All people, processes, technology must have declared and transparent levels of trust for any transaction to take place.
7. Mutual trust assurance levels must be determinable.
8. Authentication, authorization, and accountability must interoperate outside your area of control.
9. Access to data should be controlled by security attributes of the data itself.

[10] Extracted from http://www.opengroup.org/jericho/commandments_v1.2.pdf accessed August 2007

10. Data privacy (and security of any asset of sufficiently high value) requires a segregation of duties.
11. By default, data must be appropriately secured when in stored, in transit and in use.

Derived from these commandments some key principles are made, these are made in the four different phases that this paper will be discussed, namely contextual, conceptual, logical and physical. At the contextual level the Jericho Project group, especially Alina Stan, has identified some key requirements for implementing a Jericho based network. These are;

“Requirements for the Security Architecture

- *Simplicity: the architecture has to be easy to understand and manageable at the following levels: Business level, Information System level (data and applications), Technological level*
- *Flexibility, scalability, adaptability, and maintainability refer to:*
  - *The security architecture must continuously be adapted to the new requirements of the business and technical changes that reflect the de-perimeterization principle*
  - *The capacity to incorporate all the business and technical requirements and changes in the security architecture*
  - *This security architecture must lead to a flexible and maintainable Jericho enabled network*
  - *The security architecture contains independent processes and modules that are interdependent and generate exchanges based on their different functionalities*
  - *The implementation of a process or module specified in the Jericho security architecture must allow efficient maintenance, making it is easy to add or modify functionalities*
- *Holistic view of business and information systems”*[11]

The architecture objective for the Trust broker is a little bit more specific than the entire range of principles that the Jericho Forum has formulated. As stated in my previous book, *'Jericho in depth... Trust broker services',* the Trust broker uses the principles that are given in JFC number 2, 4, 5, 6, 7, 8 and 10. In addition to that, the Trust broker framework will focus on providing simplicity for the business level, which will enable – as stated in the 'Security Architecture' – flexibility, scalability, adaptability and maintainability. With these qualifications, as given above, in mind the architecture objective for the Trust broker framework is:

“The Trust broker Framework will comprise an enterprise wide architecture that is focused on the business, its processes and how to secure these. By means of this architecture insight will be created on how the information systems, aimed at enabling de-perimeterized networks can be deployed. Furthermore, it will provide the business with the flexibility and freedom in communication it demands.”

[11] Extracted from the security architecture by Alina Stan

Architecture scope

The scope of this project the scope is to create a Trust broker framework – as specified in the architecture objective - that complies to the Security Architecture formulated by the Jericho Project. Furthermore, it will provide insight into all connections it has with the other functionalities or modules within the Security Architecture. The Security Architecture is based on the same scope as the Jericho Forum. The overall scope that the Jericho Forum has defined is:

"Jericho Forum exists to develop principles and standards for secure collaboration and commerce

over open networks. The security issues of concern have two common themes:

- *They are ICT related (rather than purely business related)*
- *They span organisational and ICT domains and boundaries (rather than issues centred on individual domains under the sole control of individual organisations).*

Jericho Forum will therefore focus primarily on information flows that span organizations and individuals and how to secure and manage these across open networks. The focus will be on business to business (B2B) and business to government (B2G) flows, but not exclusively. It will take into account information flows involving for example employees, customers and the general public.

Jericho Forum will consider all aspects of security: confidentiality, integrity, and availability (some authorities treat communications security issues such as non-repudiation and privacy related issues such as anonymity as additional aspects of security; all are in scope). It will focus on business drivers as well as security topics and work collaboratively to address detailed technical requirements for these topics. It will take the vendor market, regulations, and economic factors into account."[12]

The Trust broker framework scope

As stated in the architecture objective, the Trust broker framework is an enterprise architecture that will be focused on;

- business
- business processes
- information processes

Each of these areas is influenced by the scope of this project. The scope will be: "To create the first three phases of the IAF framework, namely contextual, conceptual and logical. The physical phase will not be dealt with in a lot of detail, this because it will be different in every situation. It is therefore more important to deliver the specific requirements needed to know how such a security architecture can be built. To accomplish this I will create a technology matrix for the Trust broker framework within the physical phase".

[12] https://www.opengroup.org/jericho/vision_wp.pdf accessed july 2007

Out of scope

For the out of scope of this project I basically have the same list of demands as the Jericho Forum, their statement is;

"Jericho Forum will not seek to develop technology, general-purpose security standards, guidance or advice to cater for the broad security and business concerns that organisations have to face individually. These include: monitoring employee behaviour, filtering 'spam' email, educating and training end-users to follow internal security policies and standards, hardening COTS IT platforms against malicious attack, organising and staffing security teams, vulnerability testing, and estimating and tracking broad security costs and benefits (beyond those associated with securing collaboration and commerce). Jericho Forum does not seek to resolve wider issues of technology interoperability that do not impinge upon security, but will consider them appropriately. It has a vested interest in the standardisation of, for example, information representation, data access methods, outsourced service delivery and distributed ICT infrastructure management, because common standards in these areas imply corresponding common security standards."[13]

Architecture constraints

The Trust broker Framework architecture must be capable to use modern and legacy technologies in order to make the transition to an de-perimeterized environment work. In addition, the project must be pragmatic and must use open standards, this does not necessarily mean it has to be open source.

Architecture assumptions

The assumptions of this architecture are;

- It is legally possible to store all required information in order to determine the reputation of an entity
- It is possible to establish a large and broad enough community for making a representative reputation of a particular entity
- It is possible to create and store a reputation of an entity without breaking the entities privacy
- Contracts will ensure entities from misuse or will give a form of compensation
- Contracts will give enforceability to the overseeing entities in order to reduce risks
- It is possible to dynamically determine someone's role based on its identity

Abstract of the contextual phase

As shown in the preceding paragraphs, the contextual phase needs an answer about the '*why*' question. This is done by defining the 'Supporting input' and the 'Architectural principles'. Also, the need for a Trust broker Framework can be explained in two steps. Firstly, the need for a de-perimeterized environment is given and secondly the need for a Trust broker. The need for a de-perimeterized network is created through the challenges businesses are facing today. Businesses need to be agile, transparent, compliant and

[13] https://www.opengroup.org/jericho/vision_wp.pdf accessed july 2007

collaborative, so that they can safely grow in a internet-driven world. In order to do this, a security mechanism has to be created that can deliver these needs to businesses without increasing security risks. This security mechanism can be achieved with de-perimeterization. According to Paul Simmonds it will be implemented in four steps over the four to ten years.
Some specific business drivers for these are:

- New business models have emerged, models that are based on electronic transactions
- Wider collaboration between organizations outside the locus of control
- Reducing the time to market with new services that were acquired through merger and acquisitions

The Jericho Forum also stipulates the need for a trust management system. It claims that trust is crucial in all human interactions, and therefore also essential in electronic collaboration. Furthermore, it has stated; that trust management is complex and expensive, that de-perimeterization requires sharing information and reputation between organizations. When achieved, this will lead to a cut back on expenses. The need for a trust management system in the sense of a Trust broker, is further complemented by Capgemini. Capgemini is of the opinion that the Trust broker is the central component within a Jericho enabled network and that it will facilitate collaboration between elements on the network, regardless of their geographical location. The business drivers for a de-perimeterized environment will be advanced through the use of the architectural context that is mainly driven by the 'management framework'. This context is further clarified by the architecture objective and scope. The objective of this book is to define the Trust broker Framework and explain how it is used in an enterprise wide architecture that emphasises on the business, its process and how to secure these. The scope of this book is focused upon the first three phases of the Integrated Architecture Framework, namely the contextual, conceptual and logical.

Conceptual

The main goal of the conceptual phase is to understand what must be done by producing a physical solution and showing the vision concept of how this can be achieved. In order to do this a description of the requirements, including the non-functional will be given. Proceeding from these requirements I will show the vision concept.

Functional Requirements for a Trust broker framework

- Broker of requests
  The Trust Broker is able to handle all kinds of requests and transform these in events and triggers (i.e. automated corrective actions).
- Generates Trust Context Reports/Profiles for entities
  The reports/profiles are made based on the retrieved values for different security attributes assigned to the data, user, endpoint etc. and based on the last logs.
- Is able to perform a trustworthiness check
  This is done by an identity check and if necessary to create a contract (this can be done at a low level, like terms of agreement, but with a signature).

- Generates contracts
  If the circumstances demand for additional security controls, a contract is generated based on the business needs and requirements of the company. In case of a more official situation the contract will be based on the two companies that are dealing with each other.
- Discovery service
  Is able to select the best fitting service for the job. The discovery service is able to select a service based on the information it has about the entity. This information is compared to the obligations and policy it has to comply with.

Non-functional requirements for a Trust broker framework

- Scalability
  The Trust Broker Framework must be able to scale in order to support a wide variety of services.
- Flexibility
  The Trust Broker Framework must be able to adapt quickly to new situations e.g. quickly support a new service within the network (Mashups).
- Auditability
  All operations and results of the services executed within the Trust Broker Framework must be audited.
- Transparent
  The Trust Broker Framework must be open and transparent so any one authorized can easily trace processes and determine bottlenecks or security flaws.
- Performance
  Must be able to quickly retrieve and analyse essential information – to perform validations and monitoring, trigger events and actions for control when needed, generate trust contracts and profiles – In some cases, considerable hardware resources are needed.
- Governance and Compliance
  The Trust Broker Framework must have monitoring and control tools within in order to easily implement low level control objects that arise from the different quality control certifications.

Security requirements

- Segregation of duties
  In order to ensure that the implementation and execution of the modules or processes is not violated/abused, the rights and privileges regarding their execution must be separated.

- Trusted sources
  Each individual module/ process must use verified and trustworthy sources; even the company’s own sources must be checked.

- Secure communications
  Communications and interactions between the other modules/processes identified in Jericho Security Architecture, or between the Trust broker services and the identified modules/processes must be adequately secured, since large quantities of sensitive information will be dealt with.

Approach

To create the vision concept – that is based on these requirements – the first step will be to create a mind-map of all the functions or services the Trust broker framework has. In the next step this mind-map will be transformed into a hierarchal scheme. And finally at the end of the conceptual phase a distinction between the different types of services will be made - as is requested of IAF - that is based upon this hierarchal scheme. This mind-map gives a good impression of all the activities that must be performed within a Trust broker framework. This does not automatically means that these activities have to be preformed by the Trust broker framework, as the main idea of this framework is to combine all these functionalities to create a secure and collaborative environment. Hereafter these activities will be elaborated in a diagram. From now on, the words functionality, module and services are used a lot. To give an overview of how these have to be used I will shortly explain these concepts. Functionality describes what kind of process are performed by the framework. A module is a certain combination of functionalities, which will present an important value. A service is something that is being performed by the modules or functions. This service will give an added value to the user.

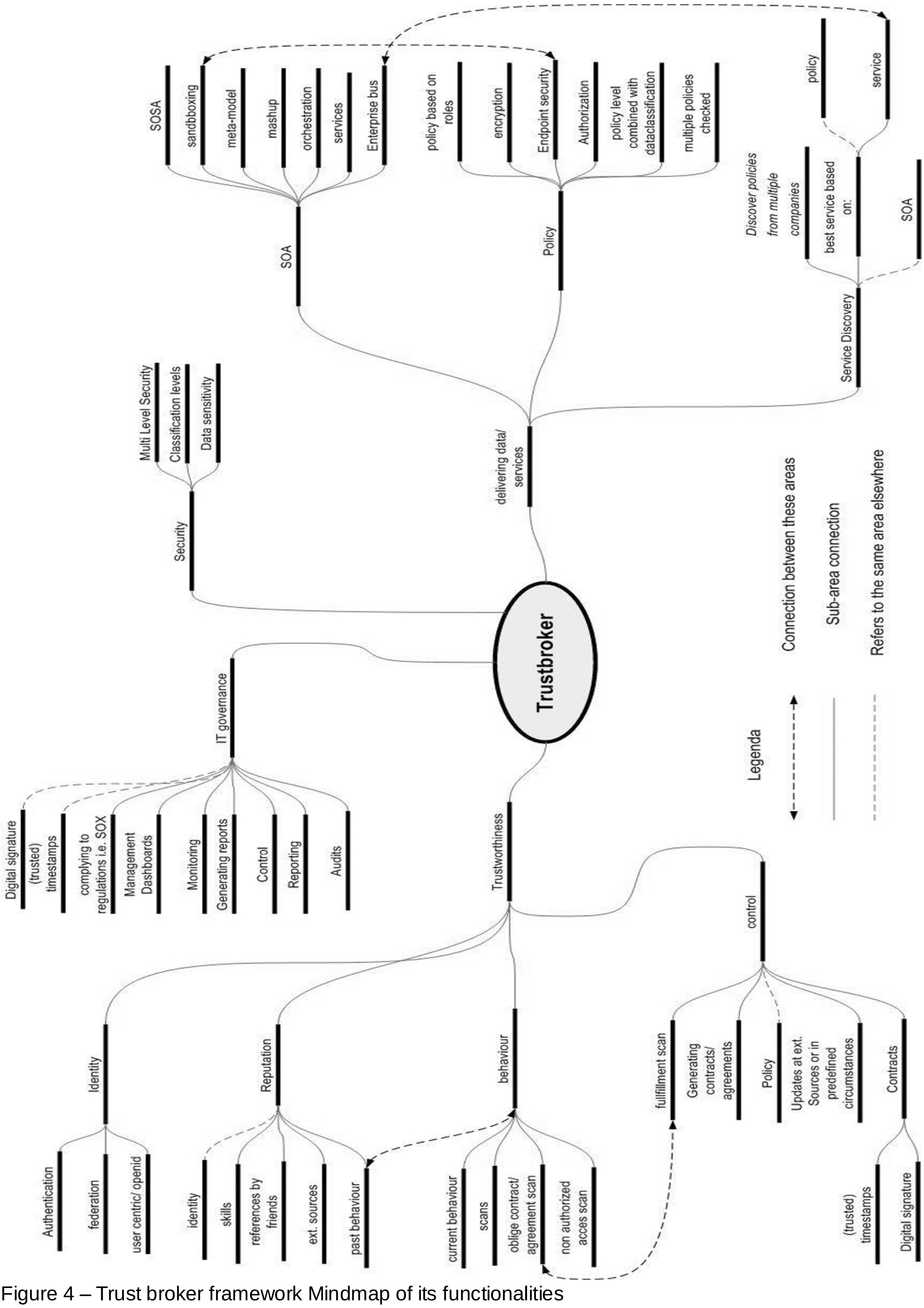


Figure 4 – Trust broker framework Mindmap of its functionalities

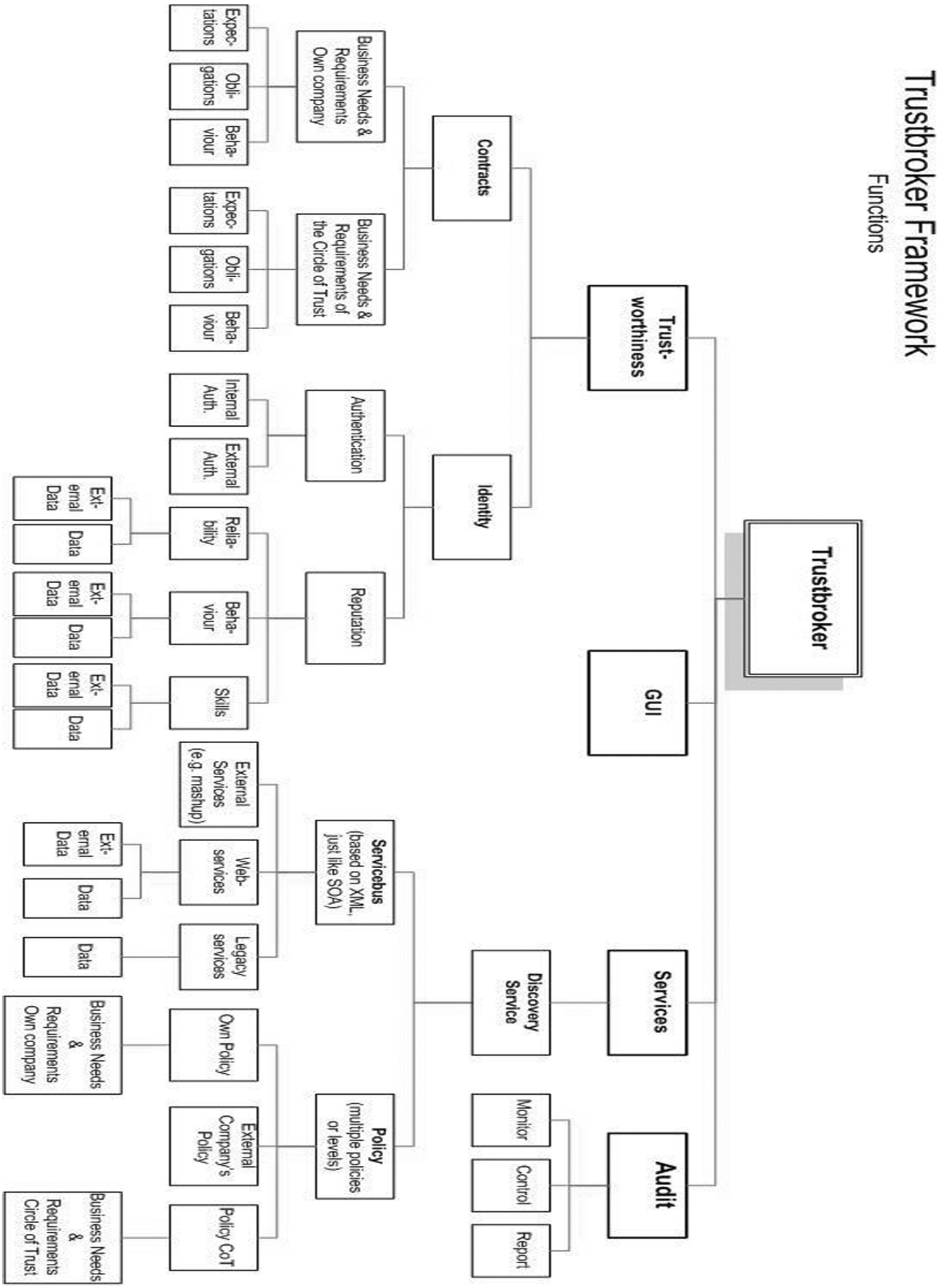


Figure - 5: Trust broker framework hierarchal scheme overview

Figure 5 continues where the previous book ended. It illustrates how the functionalities of a Trust broker framework are presented and related to each other. The Trust broker framework mainly exists out of four functions that are completed by several underlying modules. These basic modules are vital in delivering the essential Trust broker capabilities.

The most important part is Trustworthiness, it defines the context that will enable the trust management system. This is made possible by two functions, one will create the context and the other will enforce specific rules – which are provided by means of a contract – in case the trust context of an entity is not enough. The GUI offers all users a generic way in which they can control and view the results when they use a service of the Trust broker. In reality however I can imagine that this module is in fact some kind of Application Programming Interface (API), which external parties can use to tunnel the result into their own environment. The other essential part of the framework is Services. This will be highly depended on the technology that SOA provides. The connection between SOA and the Trust broker is further explained in Appendix B. Within Services two modules will make the connection to the Trust broker framework, these are the service discovery and policy. A third one the servicebus is purely technology that is enabled by SOA. Furthermore a third main function is created, Audit. In order to fulfill and comply with regulations this function is deliberately placed on a high level so that audit trails are relatively easy to effectuate. In this way the entire system can interact with the audit functionality.

A detailed description of these services is given in the next five sub-chapters.

Detail level one

Functions within each module of the Trust broker (level 1):

Module	**Function**
Trustworthiness	• Need to establish a Trust Context Report (trust level) with the use of a few Security Attributes (parameters) so it can share this information with **Services.** • Establishing trust is based on **identity** and **contracts** (but is extensible) • Trust Context Report (trust level) is defined by all modules under **Trustworthiness.** This is done by a certain algorithm (not yet defined). • This report/ level will not be stored (maybe cached for a specific time), so it can be easily adapted when circumstances change, and the SPOF is reduced (because it can not be altered at one specific point, db) • Basically the Trust Context Report (trust level) is a chance calculation. Which certainty can be given to an entities security attributes or claims?
Gui	• Displays results, data or services. • Displays this information in a highly customizable way (to the need of a entity, and maybe based on a role, trust level or policy).
Services	• Will supply all the data and services that are offered by the company or CoT. • Is extensible by adding new modules • Gives/ channels its information back to the **GUI** • Needs data from **Trustworthiness** to begin the **discovery service**.
Audit	• Displays the results from the all the audit modules that are implemented in the network. It can be compared with a management dashboard. • To accomplish this all kinds of monitoring and control tools are used. This information will be checked on whether certain requirements are met.

As said above this table presents the main four functionalities of the Trust broker framework. In general all sub functionalities/ modules are placed under these functions and report to them.

Detail level two

Functions within each module of the Trust broker (level 2):

Module	**Function**
Trustworthiness → Contracts	• Determined by the *business needs & requirements* of all relating companies. o For very important issues – i.e. a partnership between two companies - this process has to be done manually (as long as computers cannot compare different sorts of information within a certain context) • If a contract is made it will store information about: o The expectations the other party has from you o The obligations the other party has towards the other party(ies) o It will monitor the behaviour of the different parties and will compare this with the expectations & obligations. (may require some human input) • Output is an accumulation of security attributes
Trustworthiness → Identity	• Determines, based on the modules beneath it, the trustworthiness of an entity and will forward this information to **Trustworthiness.** o Trustworthiness of an entity consists basically of: ▪ The identity, thus *authentication*. ▪ The *Reputation* about the identity. • Output is an accumulation of security attributes
Services → Discovery service	• Within SOA a discovery service determines which services will best serve the needs of an entity, based on the information it has about this entity and the role in which it is acting. o This information will be received from **Trustworthiness** by means of a Trust Context Report, such as the limitations of a contract (which obligations), which identity (what skills and reputation), etc. o Information from trustworthiness can be used to determine the role off the entity. • Based on the role the corresponding policy levels will be applied.
Audit → monitor **Audit** → control **Audit** → Reporting	• Based on the specific audit process, specific information about the system and its process must be gathered. • Based on the information required the evaluation control mechanism of the audit process will be enabled. • Based on the study and testing of the evaluated controls a report (conform the guidelines of the audit process) will be generated.

These functionalities create the first modules and are the true core of the Trust broker framework. These ones are responsible to get the data that is needed to decide over the trust level of an entity.

Detail level three

Functions within each module of the Trust broker (level 3):

Module	Function
Trustworthiness → Identity → Authentication	• Determines if an entity is authentic, based on something it knows, has or is. o Can be checked through own sources or external sources (federated or user-centric) o Output is a security attribute o It is possible that different kinds of entities need other authentication methods, i.e. Companies are checked by the house of commerce, devices are checked by the manufacturer and people are checked by the stated and or a specific identity silo.
Trustworthiness → Identity → Reputation	• Determines an entities reputation, based on the modules operating under reputation. o Basic, required modules are *reliability, behaviour* and *skills*. o Output is a security attribute
Services → Discovery service → *Service bus*	• The *Service bus* will make it possible to integrate any service (through the use of open standards) in the Trust broker framework. o Must be very flexible and interoperable (support SOAP & REST approach, XML, etc.)
Services → Discovery service → *Policy (level)*	• The *Policy* module is an assembly from all the different policies that several companies or applications demand. o *Policy* uses the role, in which an identity acts, that is determined in the **Discovery Service** or established earlier in the process, i.e. with the authentication at a company. Based on this role it enables the corresponding authorization process (active authorization, restrictions and clearances).

These sub-modules describe how the main modules are being performed. They emphasize the requirement of segregation of duties, as explained in paragraph 2.3.1.3. These modules don't necessarily have to be performed at one specific geographical location or organization. This is made possible since the de-perimeterization vision creates a way to communicate free and open over the internet in a secure manner. Due to this opportunity, these modules can be performed by other companies that are specialized in that area. However, when certain modules are outsourced they will become services for your own environment.

Detail level four

Functions within each module of the Trust broker (level 4):

Module	Function
Trustworthiness → **Identity →** *Reputation* → reliability	• Reliability is determined, through own and/or external sources. The main characteristic of reliability is that this information is created by an analyses over past actions. It contains information about or like: o Financial credit, criminal record, psychological record, recommendations from other people (preferable in the same context as required, i.e. work), etc. o Output is a security attribute
Trustworthiness → **Identity →** *Reputation* → behaviour	• The behaviour of an entity is analyzed through own and/or external sources. The main characteristic of behaviour is, compared to reliability only different in time aspect, created by an real-time analyses. Analyzed behaviour information will give; o If the entity is compliant with policy and contracts, doesn't show suspicious behaviour, etc. • Output is an accumulation of security attributes
Trustworthiness → **Identity →** *Reputation* → skills	• The Skills of an entity are given through own and/or external sources. • Skills will determine what an entity can do, this can be coupled with policy to establish what the entity is allowed to do with his skills. • Output is a security attribute
Services → **Discovery service →** *Service bus* → External services	• This module will make it possible to integrate any service form a external company (as long as it supports certain requirements).
Services → **Discovery service →** *Service bus* → Web services	• This module will integrate any webservice or service presented through an enterprise/service bus into its own framework, whether from your own company or an external one.
Services → **Discovery service →** *Service bus* → Legacy services	• This module will integrate any local legacy service and if necessary will enable it to work like a webservice through the use of the servicebus.
Services → **Discovery service →** *Policy (level)* → *Own authorization proces*	• This module will create or generate (if possible) an authorization process based on the policy of your own company.
Services → **Discovery service →** *Policy (level)* → External authorization proces	• This module will incorporate the authorization process from a external company and will apply it to all roles from that specific company.
Services → **Discovery service →** *Policy (level)* → CoT authorization proces	• This module will incorporate or will create/generate (just as own authorization process) the Circle of Trust authorization process. In case of creation/generation it is based on the policy of the Circle of Trust.

Detail level 5

Functions within each module of the Trust broker (level 5):

module	**Function**
Trustworthiness **Identity** → *Reputation* → *Reliability* → External data	• Data that originated from some external source and has some specific information about the reliability of the entity in question, or considering privacy issues only information that is relevant with the role he is accessing the network. i.e. Financial credit, crime record, etc.
Trustworthiness → **Identity** → *Reputation* → *Reliability* → Internal data	• Data that originated from inside the company or Circle of Trust and has some specific information about the entity in question. i.e. Recommendations, status, etc.
Trustworthiness → **Identity** → *Reputation* → *Behaviour* → External data	• Data that originated from outside the company and has some specific behaviour knowledge about the entity. Behaviour, as stated earlier, is about current or recent actions of the entity.
Trustworthiness → **Identity** → *Reputation* → *Behaviour* → Internal data	• Data that originated from inside the company or Circle of Trust and has some specific information about the behaviour of an entity.
Trustworthiness → **Identity** → *Reputation* → *Skills* → External data	• Data that originated from outside the company and has some specific information about the skills of an entity. These skills are verified by other (recognized) companies. i.e. Schools, training institutes, etc.
Trustworthiness → **Identity** → *Reputation* → *Skills* → *Internal data*	• Data that originated from inside the company or Circle of Trust and has some specific information about the skills of an entity. These skills are verified by your own company by certifications, function, etc.
Services → **Discovery service** → *Service bus* → *Web services* → External data	• Standard output from your external company (web) service that will be transformed, if necessary, so it can be used within the local framework
Services → **Discovery service** → *Service bus* → *Web services* → Internal data	• Standard output from your own company sources that will be transformed or will be interpreted directly to work with other (web) services.
Services → **Discovery service** → *Service bus* → *Legacy services* → Internal data	• Standard output from your own company sources will be transformed or used directly into a companies legacy services.

Module (level 5)	Function (level 5)
Services → **<u>Discovery service</u>** → <u>*Policy (level)*</u> → *Own authorization process* → Own policy	• A policy is created by means of, or based on, the business needs & requirements. This process can be made convenient arranged so it can be evaluated with it-process, but it will or must be checked by human input.
Services → **<u>Discovery service</u>** → <u>*Policy (level)*</u> → *External authorization process* → External policy	• Is a means of incorporating a other companies policy (if authorization must be performed actively and is changeable over time).
Services → **<u>Discovery service</u>** → <u>*Policy (level)*</u> → *CoT authorization process* → Cot Policy	• A policy is created by means of, or based on, the business needs & requirements of the Circle of Trust. This process can be made convenient arranged so it can be evaluated with it-process, but it will or must be checked by human input.

Detail level six overview

Functions within each module of the Trust broker (level 6):

Module	**Function**
Services → **<u>Discovery service</u>** → <u>*Policy (level)*</u> → *Own authorization process* → *Own policy* → Business needs & requirements Own company	• Based on vision, mission and goals of the company the needs & requirement for the architecture, business flow and security aspects will be visualized.
Services → **<u>Discovery service</u>** → <u>*Policy (level)*</u> → *CoT authorization process* → *Cot Policy* → Business needs & requirements Circle of Trust	• Based on vision, mission and goals of the company the needs & requirement for the architecture, business flow and security aspects will be visualized.

The last two tables showed where the data will be coming from, and since the framework is operating within a circle of trust data will be supplied by many different partners. As a consequence, the information is divided into three kinds: external information, information within the circle of trust and information of one's own sources.

Categorization of the functionalities

The next step will be categorizing the activities, that I identified in the mind-map given in figure XX, according to the the five main service types of AIF. This categorization is made from the viewpoint of the system architecture, this is why some business services are categorized as an information service.

The five service types of IAF are;

1. Business Services (BS),
   A business service describes a basic building block or an activity of the things a company does. It can be understood as an single element of work that serves a single purpose. Furthermore it can provide to an internal or external party of the business, such as a manager, customer or partners. For example, within a fish company catching fish is a business service.
2. Business Information Services (BIS),
   This kind of service describes the information communication behaviour within the business. In most business services this information is necessary in order to be capable to perform its tasks, the Business information services explicitly describe those communications. I.e. of the fish company a business information service is the communication that an actor informs the company that two boxes of fish are caught and send.
3. Information Services (IS),
   The information service supports one or more automated business (information) services. This could be compared to custom-made and generic product suites. Through development these specific processes or now standardized in one service. For example, messages given to the company are done through Information Objects, such as an order list. These are dealt by an automated information system, such as an ERP system.
4. Business Object (BO)
   This is an object – often a physical object - that supports a specific business activity. In the example of the fish company this could be a fishing rod.
5. Information Object (IO)
   This is an object that only contains information which help to perform the business, such as an order list.

Main function	Main activity	Sub-activities	Service type
Trustworthiness			BS
	contracts		BS
		Businesses needs & requirements	IO
		Businesses needs & requirements	IO
		Businesses needs & requirements	IO
	Identity		BS
		Authentication	IS
		Reputation	IS
GUI			IO
Services			BS
	Discovery service		BIS
		Servicebus	BIS
		Policy	BS
Audit	Audit		BS
		Monitor	IS
		Control	IS
		Reporting	BIS
General		Trust Context Report	IO
		Security attributes	IO

In a regular architecture implementation these service types are needed in order to define the business contracts. In these contracts it becomes clear which actors and services work together, furthermore it will define to which non-functional requirements these relations have comply. However, due to the nature of this document no specific actors or relations are described, because they vary according to the different business contexts. For example, in a business that is very internet driven and which supplies many services, it is more important to quickly determine someones identity and which services an entity wants to use. But if a company operates in a static environment, only deals with specific partners and uses highly sensitive information it is far more important that policies and contracts are upheld. Furthermore, it will be important to let audit trails be performed.

This graphic represents the 'wheel of Andy Mullholland'. The main philosophy behind the use of a wheel is that all services of a company are build around the data and in order to use these data certain functionalities are required. It shows how these services have to be seen in the context of a service oriented environment.

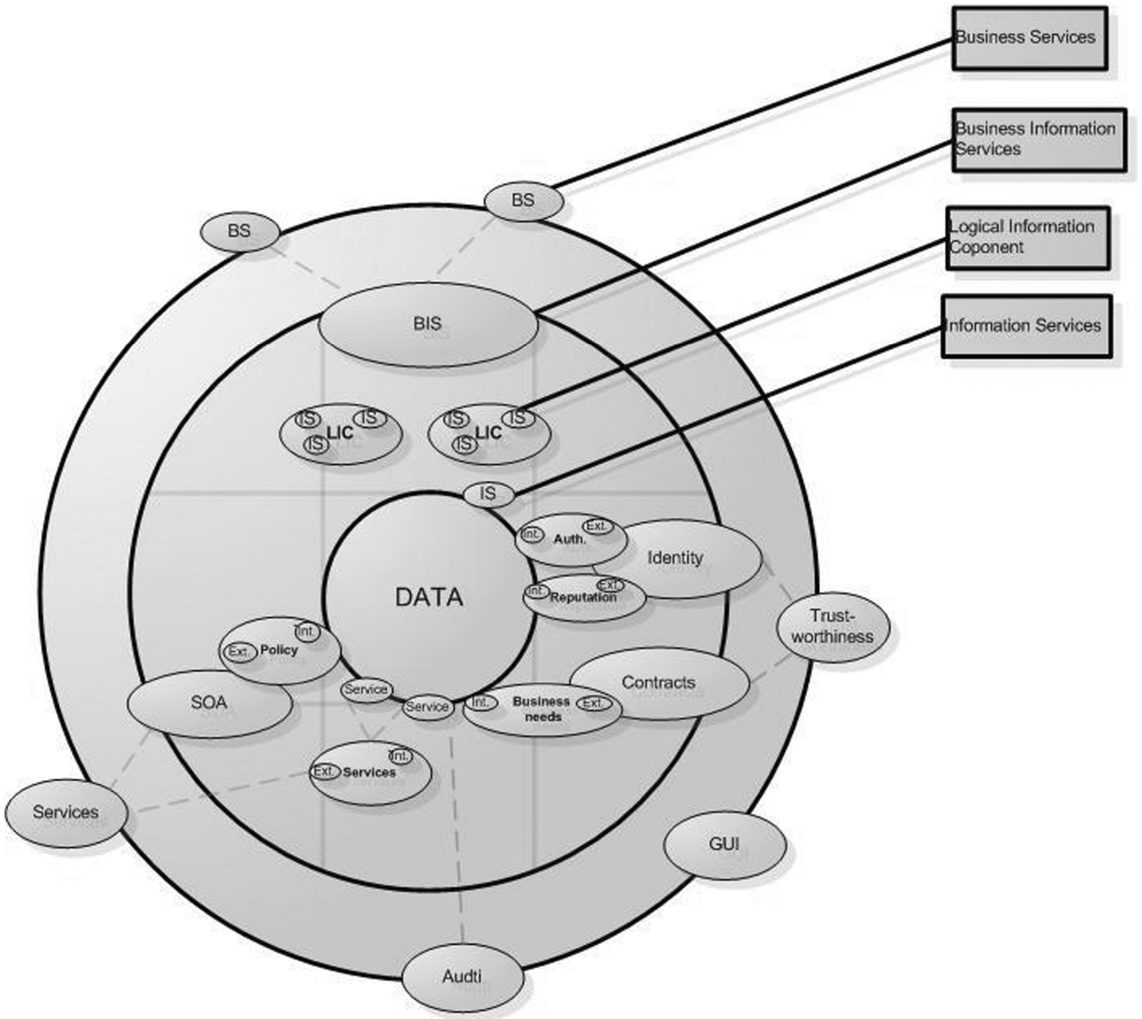


Figure 6 – The wheel of Andy Mullholland

These functionalities can be related directly to the business process, in other words the business services. But these services need data and these will be provided by the Business Information Services, which in their turn are dependent on the Information services. So the only thing the outside world will see are the business services. The other services – which make it possible to deliver these business services – are commonly know as the back office of a company.

Abstract of the conceptual phase

By means of the conceptual phase it has been explained and shown how the contextual phase can by realized. This is done by listing the functional and non-functional requirements of a Trust broker framework. With these requirements in mind an overview of a Trust broker framework is made which roughly explains all the different functionalities this framework should to have. The most important functionalities it has to provide are: trustworthiness, identity, contracts, services, service discovery and policy. With these functionalities incorporated into the architectural framework the essential part of the Trust broker framework has been completed. After which the focus can be aimed at the physical aspects of the Trust broker framework. Throughout the six tables all modules and their interactions are defined. This specifies which tasks have to be performed by certain functionalities. In combination with the requirements this can lead to specific product or vendor selection when an implementation project is started. In addition, these functionalities are coupled to one of five specified services as defined by IAF. These five services are roughly divided into two groups, the business and information services. The business services provide customized processes that are adapted to the company. The information services provide a generic solution or procedure – which is mostly executed electronically – in serving the business processes. This means that the specification of these services will show what to expect of these functions and in which manner they can be implemented within an enterprise.

Logical

The main goal of the logical phase is to define how the architecture may be structured and to describe the to-be situation. In this phase I will provide the next level of detail to the vision concept created in the conceptual phase. I will discus three different solutions guidelines that are closely related to each other.

Approach

The logical components will be created by means of the mind-map, as given in chapter 2. These logical components derive from the Business Services, Business Information Services and Information Services, but are now placed in a specific context in which they are logically ordered. For more information about the Logical components see Appendix A – methodology.

The transformation to the logical components

Before starting the transformation process from the business services to the logical components a list, of potential functionalities, modules and functions will be made output of which a selection can be made. The mind-map is used to define these functionalities, the doubles however are removed from the list.

Identity: authentication, federation, user-centric.
Reputation: skills, references, ext. sources, past behaviour,
Behaviour: current behaviour, scans, oblige to contract/ agreement scan, non-authorized access scan
Control: fullfilmentscan, generating contracts/ agreements, policy, updates at ext. sources or predefined circumstances and finally the contracts which contain: trusted timestamps and digital signature.
IT Governance: complying to regulations, management dashboard, monitoring
Policy: policy based on role, encryption, end-point security, authorization, policy combined with dataclassification, multiple policies checked.
SOA: sandboxing, meta-model, mashup, orchestration, services, enterprise servicebus.
Service Discovery: discover policies from multiple companies, SOA and determining the best service and policy.
Security: multi level security, classification levels, data sensitivity

The functionalities above are presented in an abstract and general way. In order to implement them they should be subdivided into smaller components on a more concrete level. This depends on soft- and hardware choices. Because these smaller components are very technically based, they will fall beyond the scope of this book.

Logically ordered components

The context that will be used is based on the three Trust broker models as given in the “‘*Jericho in depth... Trust broker services*’” book. The three models that are suggested in this book are; a centralized Trust broker, a client/server Trust broker and a peer-2-peer Trust broker model. These models were created with certain business models in mind and not a specific technical architecture. Therefore if you use the technical perspective to create a context you can notice a great resemblance between the three models.

For example; for the context of the Central Trust broker a distinction between whether or not a segregation of duties has to be applied can be made. With the server/client Trust broker the most obvious context that can be chosen would be between server and client. And finally for the context of the peer-2-peer Trust broker a distinction between local and general peers can be made. The choice for the final context will be based on the similarities of the three models.

The only logical similarities that can be made out between the different models are in- and outside. Inside will be everything that is under your own control, up to certain level of course. Outside is everything that is not under your direct control, but you have access to in order to use its services. Why this is the most logical decision I will clarify in the next three points:
The connection between in- and outside and the three contexts:

1. Segregation of duties or not.
   If a segregation of duties is applied to a service, it means that the responsibility of this service no longer rests with the same person(s). This concept is obviously used to increase security and liability. Segregation of duties implies that you spread control over a group of people. Alternately this could be explained as, do I keep this service inside my locus of control or outside. This segregation does not necessarily mean that the service has to be physically separated from the other services, this separation could also be accomplished on a organizational level.
2. Server / client.
   With the server/client context we do not have to look at it from a managerially perspective. This context is all about moving a service physically in or outside your perimeter. Of course the thought of a perimeter in the digital world has to disappear, but on a corporate level the separation between inside the company and outside the company will continue to exist (at least for now).
3. Local or general.
   This model is closely related to the client/server approach but is different due to the fact that each Trust broker still has some control over the general information in comparison with the client/ server approach. The most important difference between these two lies in the fact that the peer-2-peer model is designed to work within a network of multiple Trust brokers, so they can work together when necessary. This community of Trust broker peers can perform additional tasks in the field of trust management, identification, reputation, compliance, policy, etc. The quality of these fields will improve according to the increase of relevant data.

Transformation of the central Trust broker

As a short remainder of what the central Trust broker does the main activities it performs are summarized:

- Managing all the trust relationships between clients and between clients and server
- Enforces global security measures and policies to all parties it deals with
- Enforces its own security measures and policies to all its hosts
- Manages the federation of identities
- Keeps records of the reputation and behaviour of the other parties
- Monitors all the contract obligations between parties

The context for in and outside will make it clear which services – or at this point functionalities – have to be taken separately for control. In this way each inside function can be under control of just one group/person. The outside functionalities have to share their power over their function.

Outside:

Functionality	**Service type**
identity	BS
reputation	IS
updates at ext. sources or predefined circumstances	IS
behaviour	IS
control	IS
generating contracts	BIS
policy	BS
contracts	BS
trusted timestamps	IS
digital signature	IO/BO
encryption	IS
end-point security soa orchestration	IS BIS
enterprise servicebus	BIS

Each of the functionalities given above is too powerful or influential to be controlled by only one party, therefore a segregation of duties has to be applied to these services.

Inside:

Functionality	**Service type**
authentication	IS
skills	IS
references	IS
past behaviour	IS
current behaviour	IS
scans	IS
oblige to contracts	BS
non-authorized scans	IS
fulfillment scans	BIS
it governance	BS
complying	BIS
management dashboards monitoring	BS IS
generating reports	BIS
reports	BIS
audit	BS
policy	BS
policy based on a role	IS
authorization	BIS
policy combined with data classification	IS
multiple policy check	IS
sandboxing	IS
meta-model	IS
mashups	BIS
services	IS
service discovery	BIS
security	IS

The functionalities given above are all services that are important. Yet they are not as influential as the functions in the outside context. This of course doesn't mean they have to be secured less. Striking is that sub functions of identity or reputation are assigned to the inside, in stead of the outside context. This classification can be explained by the fact that the other party in control of an important process may not control everything in this process, because this would make him too powerful. So by giving control of the individual services to another group a natural equilibrium of powers is created. This will ensure that no group will become too influential.

Transformation of the client / server Trust broker

Before the transformation to the in and outside context is made a short overview of the activities that are done by the server and client is given. This will act as a remainder of what the client/server Trust broker does.

Activities dealt with by the server are:

- Managing the federation of identities
- Keeping records of the reputation and behaviour of the transactions between clients and server
- Keeping copies of all the obligations that there are between its clients
- Enforcing global security measures and policies on to the clients
- Providing a unique role with providing and storing PKI keys and certificates

Activities dealt with by the clients are:

- Keeping records of the reputation and behaviour of the transactions between clients
- Keeping records of the obligations is has with all of its clients
- Enforcing global security measures and policies to all the transactions it makes with other clients
- Enforcing its own security measures and policies to all its own hosts

For this model in and outside will mean the physical border between these functionalities. Within this context inside means the client and outside the server.

Outside context – as shown on the next page – about the functionalities that require intensive computational and disk requirements or merely functionalities that gain additional advantages if they are managed by a central unit. The inside context on the other hand will update its information to the outside and will only perform domain specific tasks.

Outside:

Functionality	Service type
identity	BS
federation	IS
reputation	IS
behaviour scans	IS
oblige to contract/ agreement scan	BS
non-authorized scan	IS
control	IS
fulfillment scan	BIS
generating contracts	BIS
multiple policy check	IS
update external data	IS
trusted timestamps	IS
digital signature	IO/BO
comply to regulations	BIS
management dashboards	BS
monitoring	IS
reporting	BIS
audit	BS
encryption	IS
end-point security	IS
authorization	BIS
discovery services	BIS
soa orchestration	BIS
enterprise servicebus	BIS
security	IS

Besides a few necessary functionalities, such as: management dashboard, monitoring, reporting, etc., it is striking to see that the server side has so many tasks. At first the main objective was to take into account that certain functionalities would gain benefit from a sort of dedicated server due to demanding technical requirements. But during the making of this table it showed that even more functionalities would benefit from access in a central environment. This last remark is the main reason why there are more tasks on the out then on the inside.

Inside:

Functionality	**Service type**
authentication	IS
reputation	IS
skills	IS
behaviour scans	IS
non-authorized scans	IS

Functionality	**Service type**
control	IS
policy based on role	IS
monitoring reporting	IS BIS
audit	BS
end-point security	IS
authorization	BIS
(soa) sandboxing,	IS
security	IS

As said above the functionalities for the inside are specially focused on the local domain in which this Trust broker acts. One may claim that the inside is inferior because it has less functions, this however is far from true. The inside acts as a kind of sensor to the outside, without the data generated by the inside the outside can't function. Furthermore, the inside is far more focused on its entities than the outside. As a result its attention is focused more towards end-point security, authorization and gathering information about the identity for establishing a reputation.

Transformation of the peer-2-peer Trust broker

In essence the peer-2-peer model has many shares many similarities with the central Trust broker. The most important difference between these two lies in the fact that the peer-2-peer model is designed to work within a network of multiple Trust brokers, which can perform certain tasks. This model is closely related to the client/server approach but is different due to the fact that each Trust broker still has some control over the general information in comparison with the client/ server approach. The central Trust broker on the other hand, is designed to work standalone.

Activities that each peer will perform are:

- Managing all the trust relationships between peers in deals with
- Enforcing global security measures and policies to all parties it deals with
- Enforcing its own security measures and policies to all its hosts
- Keeping records of the reputation and behaviour of the all the parties it deals with
- Monitoring all the contract obligations between parties it has an agreement with

Activities that the community of peers can deliver are:

- Managing the federation of identities
- Keeping copies of all the obligations that there are between peers
- Enforcing global security measures and policies on to its community of peers
- Playing a unique role in providing and storing PKI keys and certificates by dividing the information over the peers

Within the peer-2-peer model outside means the community of peers and inside only one peer.

Outside:

Functionality	Service type
identity	BS
federation	IS
oblige to contract/ agreement scan	BS
multiple policy check	IS
trusted timestamps	IS
digital signature	IO/BO
comply to regulations	BIS
encryption	IS
discovery services	BIS

A community of peers is ideally suited to perform tasks concerning very sensitive information or can profit from the shared computational power. The outside of the p2p model has much resemblance to the outside context of the server/client model, in respect to the separation criteria. However, it does differentiate with regards to the physical requirements, because the p2p model in comparison with the server/client model is not a dedicated server. As a consequence the outside functionalities are more focused on spread out data over several services or have a central unit that orchestrates the flow of information.

Inside:

Functionality	**Service type**
authentication	IS
user-centric	IS
reputation	IS
skills	IS
behaviour scans	IS
generating contracts	BIS
oblige to contract/ agreement scan	BS
update external data	IS
non-authorized scan	IS
control	IS
fulfillment scan	BIS
control	IS
policy based on role	IS
multiple policy check	IS
management dashboards	BS
monitoring	IS
reporting	BIS
audit	BS
encryption	IS
end-point security	IS
authorization	BIS
soa sandboxing	IS
discovery services	BIS
soa orchestration	BIS
enterprise servicebus	BIS
security	IS

Peer-2-peer model functions - inside context

As can been seen in the list above the inside context of the p2p model is very similar to the central Trust broker model. In essence the inside model is a customized central Trust broker, with the ability to supply certain functionalities to the community of peers.

Business Service Contracts

When the categorization of the functionalities for each Trust broker model has been completed the businesses service contracts can be formulated. These contracts determine specific requirement to a relationship between certain services, generally known as Service Level Agreements (SLA). These SLA's comply with the non-functional requirements of the architecture, but are not limited by these requirements. These contracts however are always based upon real scenarios and are dependent on the special business goals of the company. An IT architect will use these business goals and the additional non-functional requirements to set-up the business service contracts. In this book only basic relationships concerning the business service contracts shall be dealt with. To name a few:

- The identification process must leave an audit trail with each identification
- Authentication must be scalable without losing performance
- The enterprise servicebus must have a high performance

When an actual de-perimeterized architecture for an company is made, these abstract concepts are replaced by facts and figures, for example the authentication process must be capable to deal with 500 unique authentication processes.

Although the current situation does not have the necessary input to specify the business service contracts, the categorization will facilitate and speed up the future implementation. The service categorization offers an explanation what to expect from the business and technical domain.

Remaining problem: securing an orchestration

Now that the vision concept has been further worked out into logical components, another security issue presents itself.

The problem that remains - after the implementation of one of the three models described above - is how to secure a loosely coupled set of services. Of course each service has its own set of protection measures but certain combinations of services can provide data that are more sensitive than each individual piece of information. But even when a specific orchestration of services does not create data that must have a higher security clearance, it is still necessary to secure an orchestration. Because relying on the security of the service and the message handling of the Enterprise Service Bus is not sufficient. This is due to the gap in security that comes stems from with loosely-coupled services. In addition to that you have to design a spare model in case the original model gets hacked.

For example, take three servers – A, B & C – and one communication channel. Servers A, B and the communication channel are secure. This would imply that the process of communication between server A and B over communication channel one should be secure. However, during B's communication on the secure channel C may be able to intercede and impose as B, a typical case of 'a man in the middle attack'. In order to counter this an additional level of control has to be added that oversees the entire process or – in the case of SOA – an orchestration. Certainly this control can be implemented into service A, but that would reduce flexibility and increase the complexity of control at A. Therefore a separate control mechanism has to do this.

To secure a set of services – or in short, securing SOA – I have created three models, two of which are different and a third of which is a hybrid of the first two models. All three models are influenced by the SOA approach, which in this case means services are presented to an Enterprise

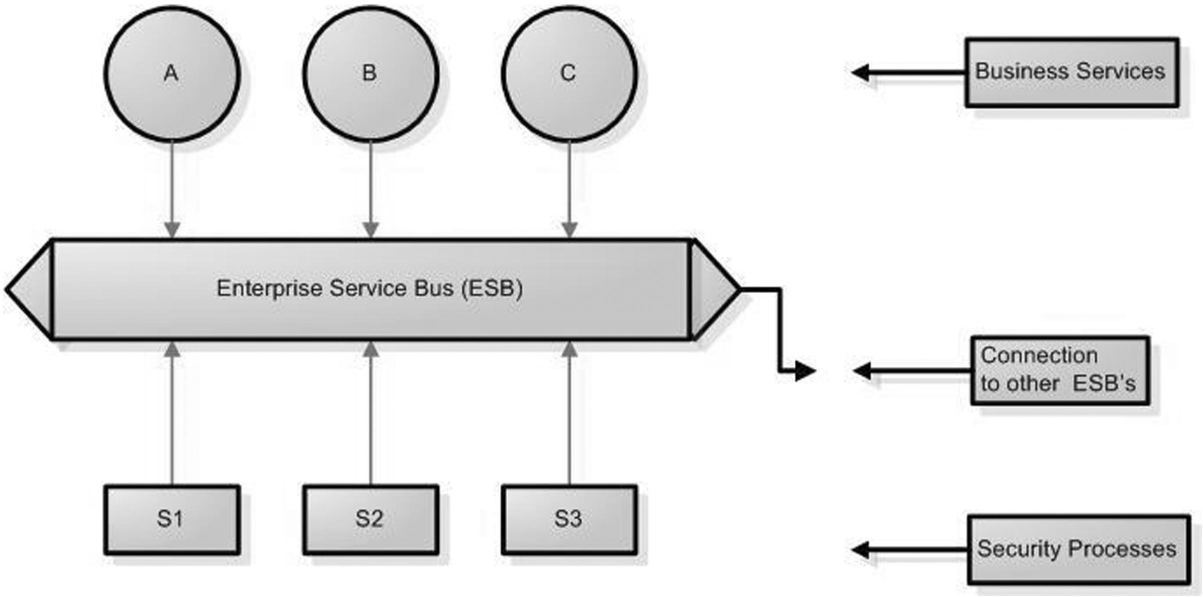


Figure 7 - Enterprise Service Bus in a SOA environment

As a critic you could say that this model will not provide security in depth, this is partially true. By securing the workflow in this manner the security will not be full proof, nevertheless it will create an additional level of security and on a higher level. So when viewed from a higher perspective these models help in creating a true strategy for security in depth.

Solution one: Secured workflow

This solution is comparable with the policy of a bar or club. To enter such a club you must go past a bouncer. Once you are inside there will be practically no additional security. Unless you want to go to the VIP room, then you will be submitted to another security process, that will be performed by another bouncer. This model is also used to a secure orchestration. A user wants to start service 'A', but to do so he is forced to go through 'security process one'. With the results from service 'A' the user wants to start service 'B', but here the user is confronted with 'security process two', etc.

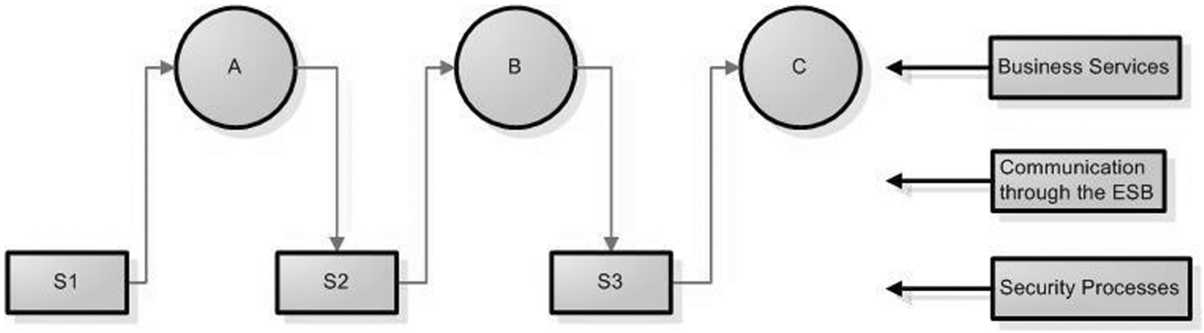


Figure 8: Secured workflow, a visual representation

Pro's and con's of a secure workflow

Pro's	**Con's**
Easy implementing. It is reasonable easy to implement, as it is a addition to the services that already exist.	Low security. Because of the great flexibility it is possible that these processes are easily by-passed or hacked, which is very bad from a security point of view. Therefore, these processes can only provide security up to a certain point, which means that high security services can not depend on this sort of options.
Flexible. Secondly it is very flexible; these security processes are just a loosely-coupled as the original services.	

Solution two: (inherent) secure service

The same analogy of the bouncer can be used for this solution. The difference between these two models is that with solution one the security process of checking people at the entrance is performed by a bouncer. But with this solution the bouncer not only performs the additional security process, he will execute the entire process. As a result the bouncer will constantly be checking whether or not the actions being performed are still in accordance with all security policies. The IT processes can be secured in the same kind of way.

In order to accomplish this model a service will start within a certain security context. This context could be compared with a sandboxing model that is used with virtual machines. Within this newly created sandbox almost anything is allowed, but in order to store information – or export information out of your sandbox – an entity has to get special authorization.

The dotted line indicates that the security context can change over time, it does not refer to different contexts.

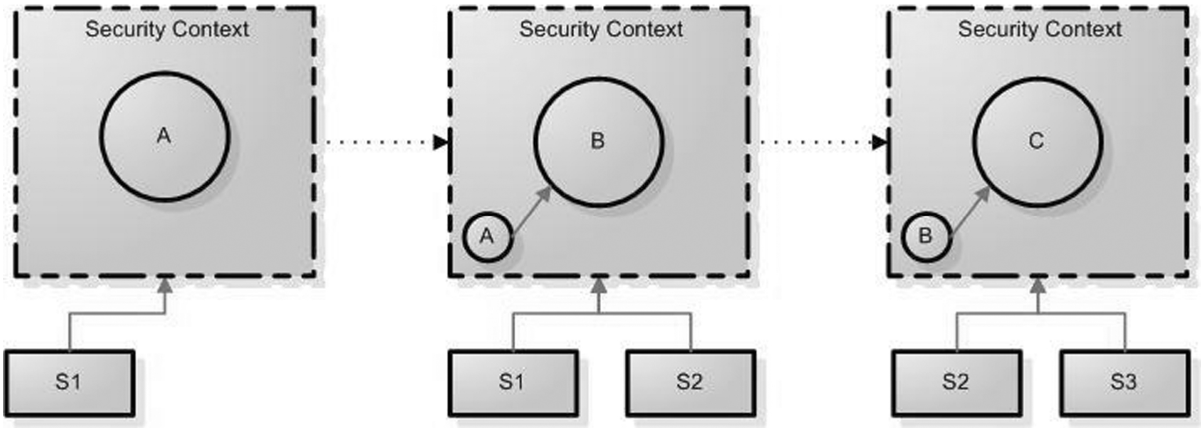


Figure 9: (inherent) secure service

Pro's en cons of the secure service

Pro's	**Con's**
Highly secure This model has the potential to create a very secure environment, this is due to the fact that it can use virtualization techniques. These techniques provide the means to easily create an environment that is totally customized to the needs of the services or circumstances.	High requirements on hardware Besides the much higher hardware demands of creating huge numbers of security context boxes
	Difficult for the accounting process. A sandbox will make it more difficult to audit. This problem is due to the fact that a lot of information is lost when a security context is closed. There are however solutions to this problem.
	High security threats for virtualization. Another problem with this virtualization technique is that it is relatively new in the business market. As a consequence, the focus has not always been on security. Presently there are some techniques - described by Joanna Rutkowska - that infect the hypervisor. So with regards to security, virtualization has to mature.

Solution three: hybrid of secure work flow & secure services
As an alternative and – in my opinion - a more realistic approach this solution is an hybrid of the first two options.

Not every service that needs security has the need for the creation of a separate security context. This would be cumbersome, expensive and in some cases it could be 'over the top'. However imagine the following; two types of information are public, but put together they may become top secret. In these sort environments a hybrid of secure workflow and secure services can be the answer.

An elaboration of the example with the hybrid model

Imagine that a bank needs to know which pin codes correspond with which bank numbers – not that a bank is allowed to do this, but it is a hypothetical question. According to figure 10 this bank will start to index all the pin codes, as this is potential sensitive information it must be dealt with accordingly. The second process, accumulating all bank numbers, is essentially more public information - at least in this case. To create a special security context for this

.

process would be a waste of resources; therefore the secure workflow approach will be used here. For example, to accumulate all bank numbers someone must be authorized. This authorization process can be 'security process 2'. At this point the system has both the pin codes and the bank numbers. Combining them may give access to the highly sensitive and classified information. Therefore, these kinds of sensitive processes have to be performed within a customized security context.

This hybrid model naturally has the benefits and disadvantages of both models, but in addition it has the following characteristics due to the combination of services.

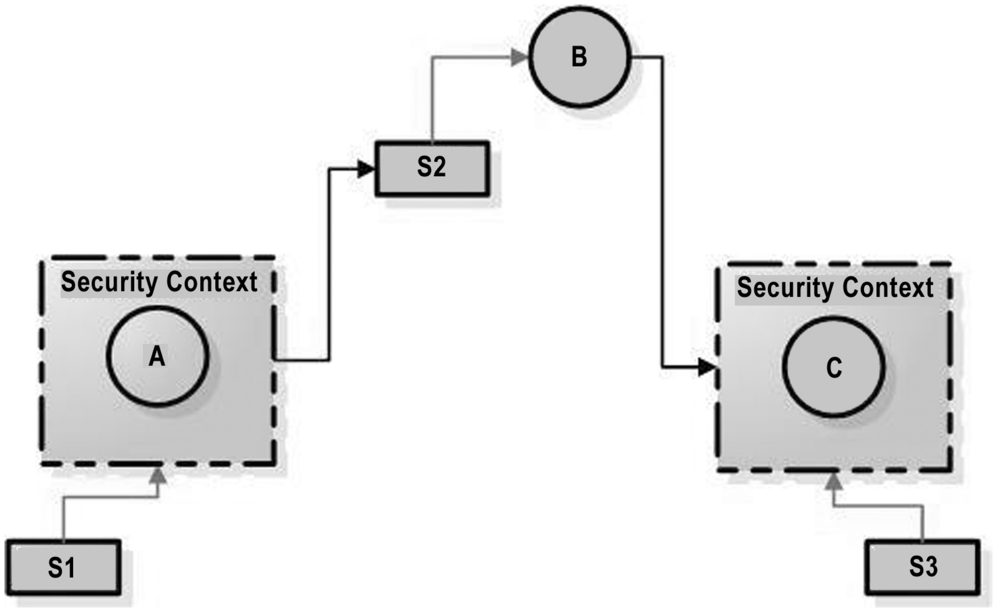


Figure 10: Hybrid of secure work flow & secure services

The pro's con's of the hybrid model

Pro's	**Con's**
Very flexible. Can use all kinds of services from different sources and secure the workflow.	Weakened Security. Although certain processes do not need high security contexts the entire workflow is weakened by using the secured workflow model. “*You are as secure as your weakest link*”.
Interoperable. Can work with different types of security mechanisms	

Abstract of the logical phase

The logical phase has to define a to-be situation by describing how the architecture can be structured. This is done by further explaining the concept vision as created in the conceptual phase. The concept vision has lead to the notion that the Trust broker framework had to be split up into four main modules, because of the need to segregated the duties. This is presented in the logical phase by using the three Trust broker models, as presented in *'Jericho in depth... Trust broker services'*, and separate them in two different contexts. The context that is used is in and outside. These contexts will be filled in with the components as defined the conceptual layer. These components are based on the main functionalities and further detailed by possible functionalities as given in the mind map. After this classification one sees what the different models have in common or what differentiates them from each other. When applied to a real scenario, services will be more specific and relations between services will become clear. This information will make it a lot easier for a company to choose which technology it has to use. Furthermore, it will provide the business with additional insight into interactions between services and contexts.

Physical

This phase will not be dealt with in much detail due to the fact that each physical implementation in a company will need other requirements. I will explain which technologies can be used to deliver the to-be situation; these will be based on the functional and non-functional as given in the Logical phase.

Technology matrix for the Trust broker Framework

This technology matrix – as shown on the next page – is based on the technologies as suggested in *'Jericho in depth... Trust broker services'* chapter 10. Although the Trust broker uses many services, not all of them belong to the Trust broker framework itself. Therefore, I will only discuss the most necessary functionalities and their corresponding technologies.

In the conclusion of *'Jericho in depth... Trust broker services'* I said that there are enough technologies to create an initial Trust broker, but not all technologies were mature enough. Within this technology matrix I will include a readiness level that is based on ten demands, this is worked out in 'Appendix D– Technology Matrix'.

Primary Function	**Functionality**	**Technology**	**Readiness grade**
Trustworthiness	Identity	OpenID	65
		MS Cardspace	60
		Higgins	65
		XRI	55
		SAML 2.0	90
	Reputation	Jyte	41
		Ebay	40
		Experian	40
		Trustplus	36
	Behaviour	Lot of information can be gathered from monitoring tools from the Audit or End-point security functionalities.	
	Contracts	Link contracts (XDI)	75
		WS-Agreement	65

Primary Function	**Functionality**	**Technology**	**Readiness grade**
Services	Policy	WS-Policy	75
		XACML	90
	SOA/ servicebus	WSDL	90
		WS-Choreography	80
	Discovery service	WS-discovery	70
General		WS-Trust	85
		WS-Security (WSS)	85
		WS-Federation	90
		ID-WSF	90
		ID-FF	90
		SOAP	90

Technology Matrix

The table above underpins the conclusion as given in my previous book, '*Jericho in depth... Trust broker services*'. It shows that there are many technologies to supply these functionalities, as given in the logical phase, but they are not yet implemented in actual products or the technology does not have the proper maturity level for implementing in a critical business environment.

Abstract of the physical phase

In the fourth phase of IAF offers an explanation with what the logical components can be loaded. Therefore, only the general technologies are presented which are able to deliver the main functionalities of the Trust broker framework.

In this phase a readiness level of these technologies is given from a business point of view. This point of view is chosen based on of the conclusion that was made in my previous book, that the technology isn't mature enough. With this in mind a company does not want to invest in technology with a bleak future. For this reason ten questions have been formulated that can give an overview of the chances that the technology will be supported in the future. Subsequently a grade is given to demonstrate the readiness level.

Information about this level will help companies in selecting soft and hardware to which they must comply, in order to make use of functionalities as presented in the conceptual phase.

Scenarios

By means of the scenarios the processes of the Trust broker Framework will be further explained, through the use of a real life example and a sequence diagram.

These scenarios are based on fictional people and companies. Furthermore no particular Trust broker model is used in these scenarios, but rather the concept of what the Trust broker framework can do.

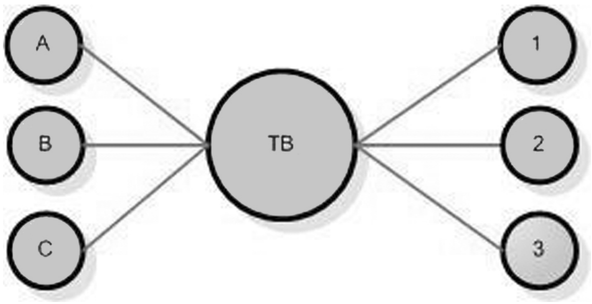


Figure 11: Possible scenarios

The graphic above offers a mathematical approach for creating the scenario's. In this model it is assumed that the Trust broker will be the constant in the center of the equation and that two variables on each side will influence different scenario's. These two variables are;

- the user and his authorization level, listed as A, B and C
- the data and its security clearance, listed as 1, 2 and 3

This approach will create 9 slightly different scenarios, but on the whole 6 of them are very comparable. The three that stand out in difference or importance will be described. In short these scenarios are;

- A – TB – 1
  The user has a low authorization level and wants to access data with a low security clearance, a.k.a. public data.
- B – TB – 3
  The user has a medium authorization level and wants to access high security data.
- C – TB – 2
  A highly authorized user wants to store data with a medium security level.

The three scenarios, as given above, will be described in the next three sub chapters. These descriptions will mainly discuss the functionalities from the viewpoint of the user. Each of these scenarios is coupled to the business scenarios as presented in the vision paper of the Jericho Forum[14]. This link between the scenarios is given by means of a fictional company, Always Secure inc.

Always Secure inc.

Introduction to the scenario script, which the three different scenarios will use. The company where all the users in this scenario work is Always Secure inc. This company finds it really important to give their employees a certain level of freedom to perform their duties. Within this philosophy the concept of flexible workspaces en teleworking fits perfectly. Additionally Always Secure inc. wants to offer communication and collaboration in every possible way to their employees, partners and customers. However, to maximize the benefits of this freedom in communication, employees and partners have to be able to access potentially sensitive company information. In order to achieve this, Always Secure inc. needed a security solution that could deliver these requirements. As a solution they chose to implement a concept that is called de-perimeterization. During the implementation Always Secure inc. has divided their employees and data into three different classes; employees were graded into secure (A), high secure (B) and top secret (C). Data was categorized into low sensitivity (1), sensitive (2), highly sensitive (3).

Scenario one: A – TB – 1

The scenario that is described here will assume that a normal user – who is graded as a secure person – will be accessing sensitive information through the Trust broker.

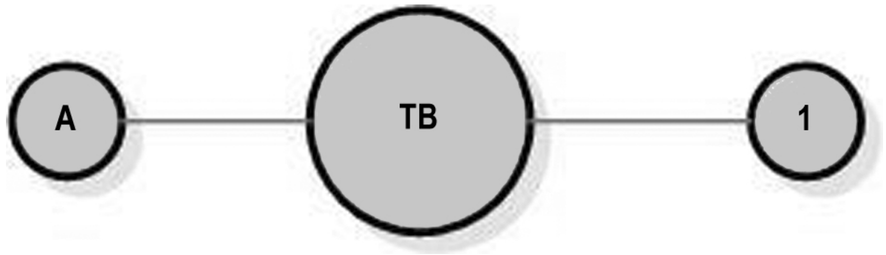


Figure 12: Scenario one

[14] https://www.opengroup.org/jericho/vision_wp.pdf accessed jun. 2007

Meet Anton. Anton is a salesman at Always Secure inc. Currently Anton is 'on the road' busy to persuade a customer to buy a product of Always Secure inc., but Anton can sense some uncertainty in the behaviour of his customer. It seems that the customer isn't sure if the product is supported good enough. To win some confidence Anton wants to show some inside information. The information is not highly secure, but not everybody needs to know it at the moment. This specific piece of information is classified as 'sensitive' by Always Secure inc.

To access sensitive information Anton - outside the locus of Always Secure inc. - needs to present some information to Always Secure inc. to confirm his true identity. To do this Anton remotely connects to the central Trust broker of Always Secure inc. to access the services he wants, or in this case the data he wants. To increase the security risk even more Anton uses the computer of his client, this because the client has a bigger screen which will come in handy when viewing some slides and pictures. If Anton wants to use the services of the Trust broker he must first identify himself. This is done with a strong authentication process. After successful identification of Anton the Trust broker will determine which role Anton has at Always Secure inc. and whether or not he may access the data that he requested based on the policy levels that correspond with his role.

Now Anton is defined as a Salesman at Always Secure inc. with the clearance level A. The data that Anton has tried to retrieve is graded with security clearance 1. This means that Anton can view this data without any problem. All of these processes are transparent to the user, the only thing Anton had to do is give the required information for the strong authentication process. And in case something does go wrong, like an identity theft or a man in the middle attack, additional processes have to be executed in order to comply with the security policies. To achieve this new detection systems have to be designed that are capable of alerting either side of this security breach. With all processes completed successfully Anton has now clearance to view the files he wanted to access. These files are some statistics of new research that has been done within Always Secure inc. to update their products. But this is not the main reason Anton wanted to show this to this potentially new client. These files also show something else that is even more important; this update was made in close collaboration with some partners. Anton thinks he can surely persuade this customer with this news. The last few weeks Anton has had some issues with his executive about the collaboration with a partner. This collaboration was created in order to adept and improve products that are already being sold, however Always Secure inc. is not entirely sure about it market readiness. Yet Anton immediately wanted to use this fact to win over even more customers, because it states that Always Secure is developing future products that are more reliable. Unfortunately, the executive of Always Secure inc. is not yet sure whether the test will be a success. If not, it would endanger the continuation of this product. As a consequence, he did not want to promote the new product yet. After an additional thirty minutes, Anton had finally persuaded this client that the product had very good support and that Always Secure inc. will continue to innovate their processes and to update its products.

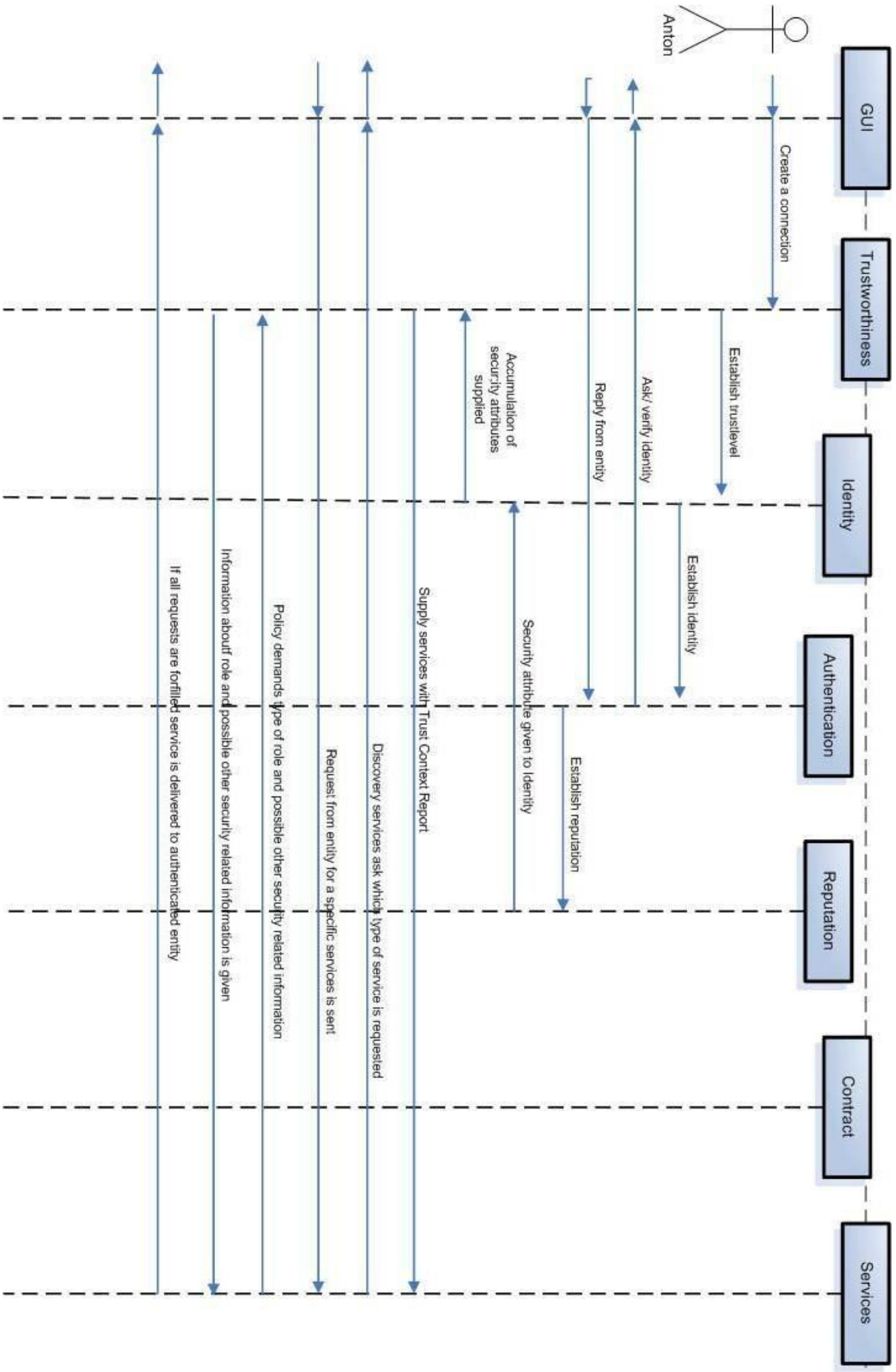

GUI
Trustworthiness
Identity
Authentication
Reputation
Contract
Services
Anton
Create a connection
Establish trustlevel
Establish identity
Ask/ verify identity
Reply from entity
Establish reputation
Security attribute given to Identity
Accumulation of security attributes supplied
Supply services with Trust Context Report
Discovery services ask which type of service is requested
Request from entity for a specific services is sent
Policy demands type of role and possible other security related information
Information aboutf role and possible other security related information is given
If all requests are forfilled service is delivered to authenticated entity

Figure 13: Sequence diagram of scenario A-TB-1

Connection to Jericho Forum

This scenario is a perfect example of a 'support roaming personnel' business scenario as given by the Jericho Forum. It specifically addresses the problem given in the context of "*Phoning home from a hostile environment".* The context that is given here is:

"Roaming personnel wish to log into the organisation's intranet to access internal applications and information from a customer's IT system or other 'foreign' remote computers."

The other context this scenario supports is that of *"enable portability of identities and data"*. The context that this business scenario gives is:

"Secure data portability may be desirable in some situations. This includes the ability to use a home computer, the corporate laptop or other arbitrary computers to work on sensitive data, with easy transfer and backup of information to the 'home' organisational server.

Roaming personnel may require portability of authentication credentials, potentially including cryptographic keys, biometric data, passwords and other relevant information. This enables use of Internet cafés and other public IT facilities, relying on a phone/ personal digital assistant (PDA) for high priority messages and data, therefore reducing the need to carry a larger portable computer."

Even though the ease of resolution of these business scenarios are not dealt with in technical detail, it does show that the Trust broker function is indeed a perfect candidate to facilitate all these technical implications through the use of services and open standards.

Scenario two: B – TB – 3

In this scenario the user has a medium authorization level and wants to access high security data.

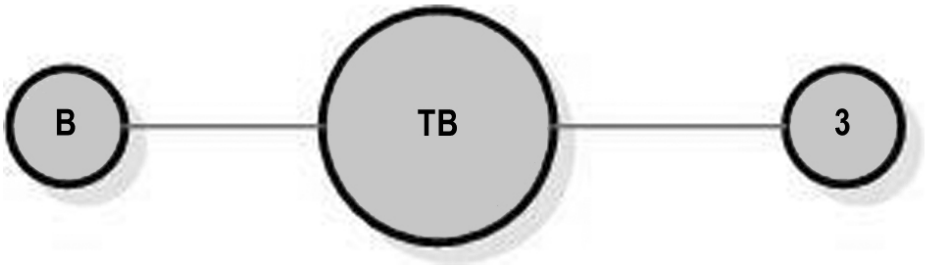


Figure 14 - Scenario two

Meet John. John is a consultant at Always Secure inc. At the moment John is a lucky man, because he has been offered a rare opportunity. He has been asked by the government to work on a top secret job. Due to the high security he has to sign a lot of special contracts in order to ensure he will not talk. Of course John will not talk because this job could give him the promotion he was eagerly expecting, and secondly he doesn't want a lawsuit or even worse, go to jail. During his job he is stationed in a high secure governmental building, from here he has access to the internet. In order to complete a document John has to access a special application of this department of the government. John establishes a connection with the Trust broker of Always Secure inc. - through which he is authenticated and authorized to work on the network of the government. Due to an agreement between the government and Always Secure inc. this department of the government is added in the Circle of Trust of the Trust broker of Always Secure inc. This implies that a number of policies and security issues are more interconnected or tuned to each other. This does not imply that these issues have become less rigid. Furthermore, these two parties can now exchange identities based on federated identity, and perhaps in the nearby future this will be done by the user-centric approach. After this connection with the Trust broker has been made, John is challenged in two ways to authenticate himself to this Trust broker. Firstly John needs to fill in a username and secondly he needs a physical token to enter a password. These challenges are called a two factor authentication, because the user is tested on something he knows and something he has.

After the successful identification, the Trust broker validates John. Now, he can access the services he needs. After he has been acknowledged by the service, another control mechanism is enabled.

This mechanism will establish three things:

1. the role in which the identity John, consultant at Always Secure inc. is acting
2. the policy this role and service has to comply with
3. checks whether or not these rules are fulfilled

When this mechanism is successfully completed the service will be presented through the servicebus to the GUI, and thus the entity

The service process gives John the role of “highly sensitive project - member” - this in comparison with Always Secure inc. would be medium level – this means that during a highly secret project John is only a member. The policy for this role however specifies that nobody with a lower clearance level may access this data. So now John is presented an error message with “Unauthorized access”, fortunately John is also presented a 'what to do' list to resolve this error message. Some options that are presented are:

- get another identity
- get an higher security clearance
- etc.

He clicks on 'get a higher security clearance', as a result he is given the option to increase his security level with a contract. Fortunately, the contracts he had signed earlier – in getting the job – are already recognized and processed by the system. So now John can access the higher security data with a upgraded medium authorization level.

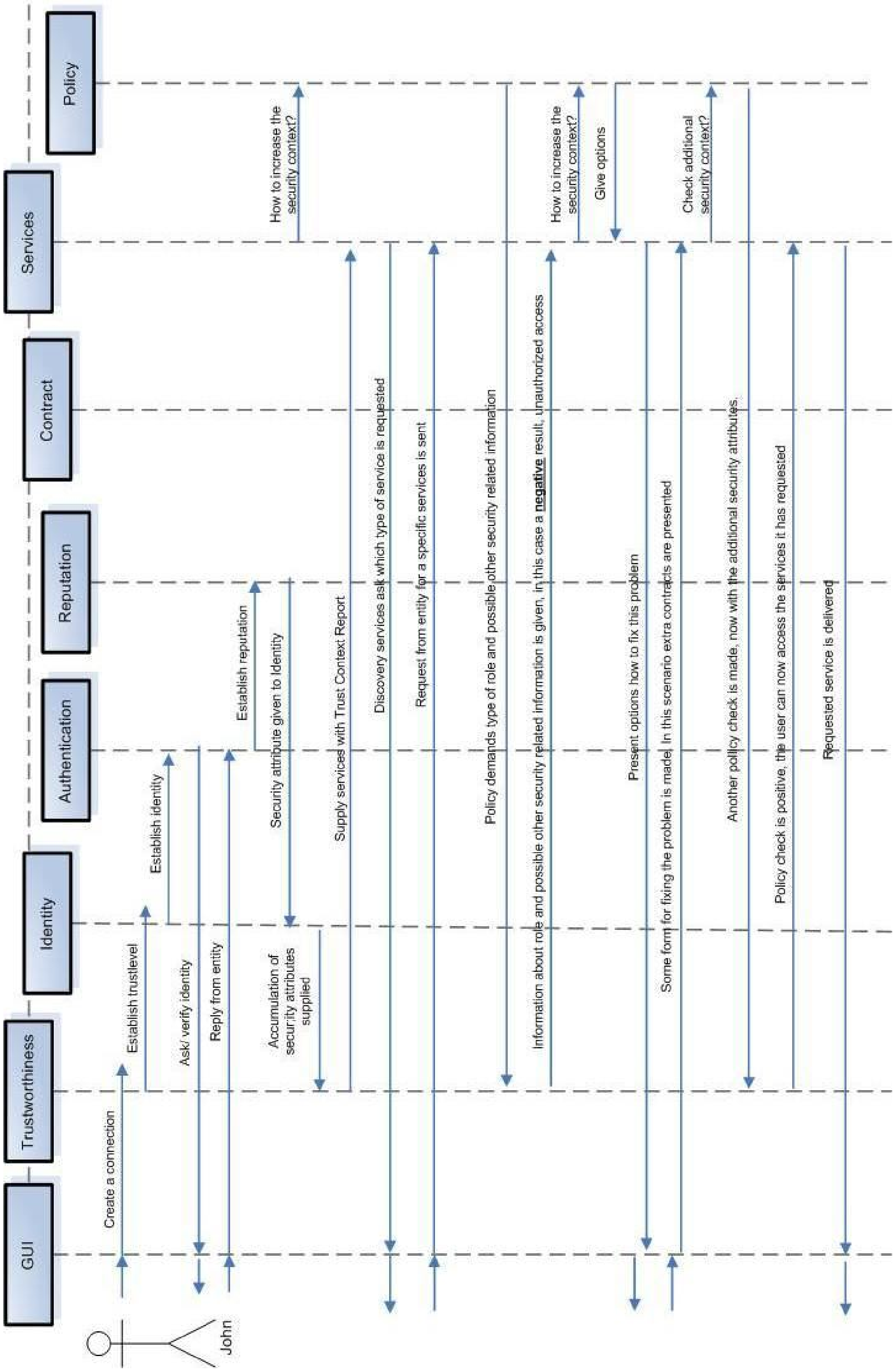

GUI
Trustworthiness
Identity
Authentication
Reputation
Contract
Services
Policy
John
Create a connection
Establish trustlevel
Establish identity
Ask/ verify identity
Reply from entity
Establish reputation
Accumulation of security attributes supplied
Security attribute given to Identity
Supply services with Trust Context Report
How to increase the security context?
Discovery services ask which type of service is requested
Request from entity for a specific services is sent
Policy demands type of role and possible other security related information
Information about role and possible other security related information is given, in this case a negative result, unauthorized access
How to increase the security context?
Give options
Present options how to fix this problem
Some form for fixing the problem is made. In this scenario extra contracts are presented
Check additional security context?
Another pollicy check is made, now with the additional security attributes.
Policy check is positive, the user can now access the services it has requested
Requested service is delivered

Figure 15 - Sequence diagram of scenario B-TB-3

Processes that are used to deliver the functionality as described above are presented in the statement diagram in 'appendix C– diagrams'. In addition a sequence diagram is given in which the events of this scenario are dealt with.

Connection to Jericho Forum

This scenario is an example of two business scenarios, namely 'allow external access' and 'provide low-cost secure connectivity'. The contexts that - with the help of John – are dealt with are:

- *Application access by suppliers, distribution agents or business partners.*
  Major applications support critical business processes including sourcing, production scheduling, selling, and purchasing. As originally designed, the applications help internal users within the organisation carry out these activities, with separate IT or paper interfaces to the external organisations involved.
- *Access over wireless and public networks*
  Organisations wish to use public and private wireless (IEEE 802.11 standards based) networks to open up access to internal systems, data and applications. The path of access may be via an Internet access point at home, or in a hotel, or via wireless networks and service providers at, for example, airports and customer/ business partner sites. Web based applications may also be deployed for public access at Internet cafés and similar facilities.

Just like the A-TB-1 scenario, this scenario also does not offer the final answer to the ease of resolution as given by the Jericho Forum. However, it does show that the Trust broker framework can be used to deliver these services. Because it creates a trusted environment in which the control and enforcement of policies are increased in such a way that partners – within the Circle of Trust – can perform these security intensive processes.

Scenario three: C – TB - 2

A highly authorized user wants to store data with a medium security level.

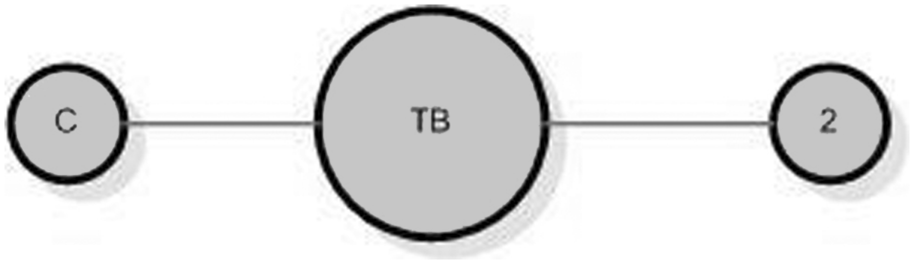


Figure 16 - Scenario three

Meet Sarah. Sarah is the CEO of Always Secure inc. She is currently busy in defining a new strategy for Always Secure inc. its future. This includes some plans to takeover another big company that will enable Always Secure inc. to acquire some long-term assignments with the government and especially the army, which will significantly enhance the security in the future. Currently Sarah is in an advanced stage of negotiations. All the arrangements are on paper and she is making the final adjustments. After working all day with documents she wants to get the opinion from someone she highly trusts. However given the nature of this document it is highly confidential information – because it can influence the stock market price – so the document is given a high authorization clearance. This doesn't necessarily mean that Sarah can't share it with anyone, but in order to do it there are high policy demands. *Giving documents a clearance level is done by two services within the Trust broker framework. One process is responsible for classifying all information, this is done by data classification. The other process depends on the policy service. In the case of Always Secure inc. the policy states that one with certain clearance level can not decrease the clearance level of a document.* Unfortunately the person of whom Sarah wants to get an opinion from doesn't work for Always Secure inc. This means that she can not send her document to him as it is to sensitive. But as headstrong as she is – something that made her CEO of Always Secure inc. - Sarah thinks of a possibility to send the document another way. Sarah thinks that changing the clearance level of the document will do the trick. So after being signed in all day at their local Trust broker Sarah wants to transform her high security document to something less important so she will be able to send it to her trusted friend for a review. But when Sarah tries to execute this change she is confronted with an error message, *'unable to save this document with to a lower security clearance'*. *(This sort of message is possible due to automated data classification and Multi Level Security (MLS) as described in the Security Architecture by Alina Stan.)* After a few other attempts Sarah gives it up, at least with the system. Sarah schedules an appointment with her trusted friend to discus the document over lunch.

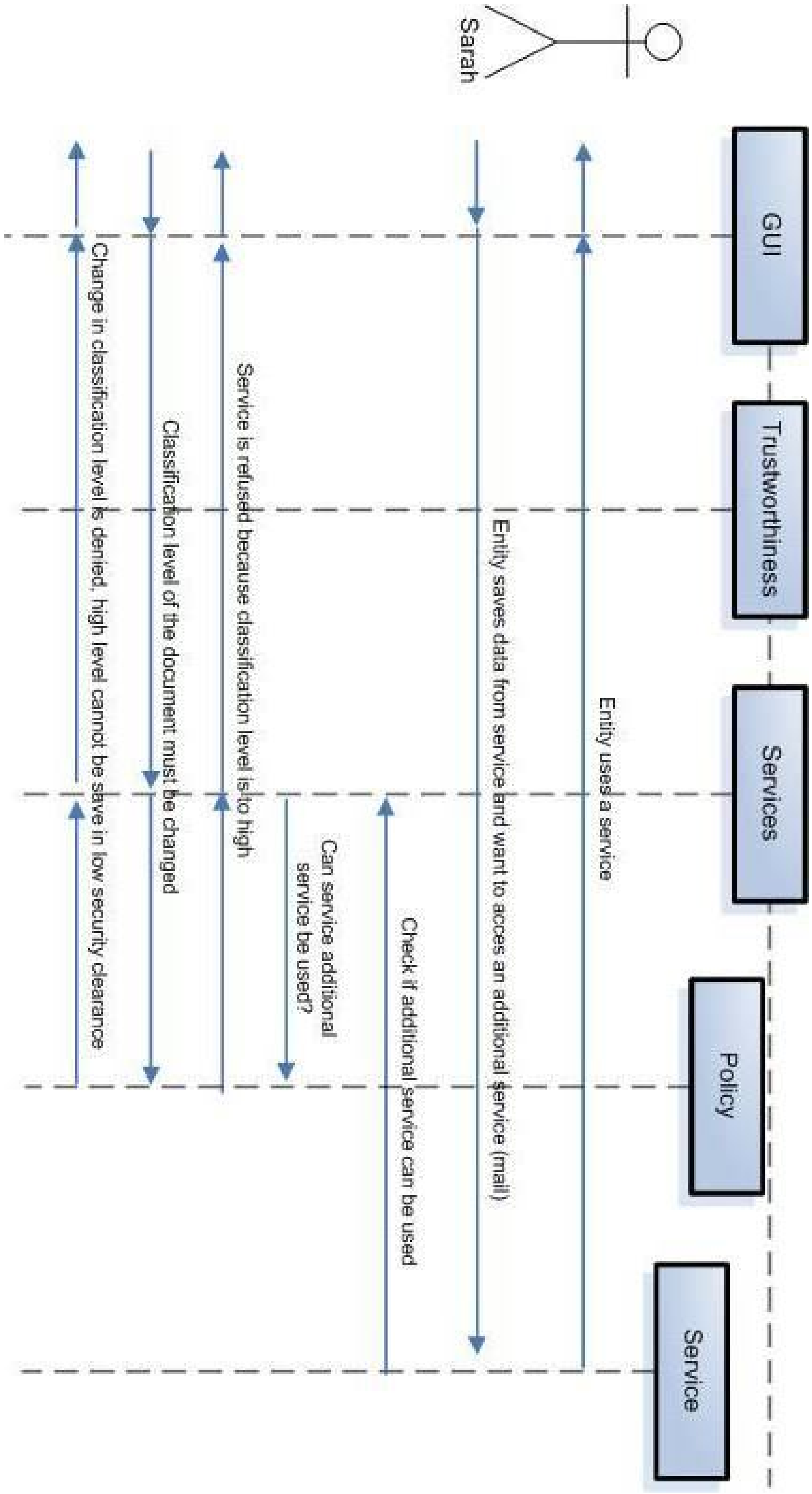

Sarah
GUI
Trustworthiness
Services
Policy
Service
Entity uses a service
Entity saves data from service and want to acces an additional service (mail)
Check if additional service can be used
Can service additional service be used?
Service is refused because classification level is to high
Classification level of the document must be changed
Change in classification level is denied, high level cannot be save in low security clearance

Figure 17: Sequence diagram C-TB-2 (authentication part is skipped, as this already happened earlier that day.)

Connection to Jericho Forum

This last scenario is an example of the 'improve flexibility' business scenarios. The contexts that are dealt with in this short scenario, are:

- Consolidate identity and access management (IAM) systems for collaboration and commerce
    - Identity and Access Management (IAM) refers to a class of security functionality and systems concerning the unified management of user authentication, authorisation, and access to data and systems. Although IAM may apply narrowly to password synchronisation and single sign-on, in its broad definition it typic ally includes issuing and revocation of common user identities and access rights automatically, and ensuring that all systems and applications in an organisation recognise these.
    - IAM systems can link to Human Resources applications and organisation wide directories so that when personnel move jobs or change roles, the IAM system determines by means of automated rules how to adjust the individual's access rights in relevant applications and systems. Directory interoperability is therefore a key issue in IAM design.
- Automate policy for controlled information sharing with other organizations
    - Organisations define fundamental information security policies (to varying degrees of formality) in terms of the information classifications involved. A classification expresses the sensitivity and/or criticality of a particular type of information, and links to the organisation's view of the risks associated with compromising its security, and the control environment that the organisation expects will apply to that information.
    - Classifications guide the evolution of organisational baselines for security controls and determine the norms for risk analysis and management. The fundamental reason to classify information is to ensure that there are organisation-wide rules for information security rather than leaving this up to the discretion of individuals.
    - So long as information remains inside the organisation, personnel may only apply classifications loosely. However, as external access and disclosure requirements increase, classifications will provide an important method to define and enforce rules and constraints to control information sharing with business partners, suppliers and customers.

Although the scenario is short it deals with some high implicating topics, many of these issues are not directly related to the Trust broker. However the Trust broker does facilitate the services required to ensure policies and secure collaboration between parties.

3. Conclusion

The purpose of this book was to elaborate on the idea of the Trust broker framework, as it was created in the book *'Jericho in depth... Trust broker services'*. This was established by means of the Integrated Architecture Framework (IAF). Through the use of the IAF phases, the Trust broker framework is made applicable. The emphasis lays on the last three phases of IAF, which respectively are conceptual, logical and physical.

In the conceptual phase the main idea of the Trust broker framework is repeated and further elaborated by means of a hierarchal scheme. This scheme shows which functionalities are needed for the framework in oder to let the Trust broker function. This scheme shows detailed characteristics of the functionalities on each layer. Based on the previous book the functional and non-functional requirements of the Trust broker Framework are given. With the combination of these characteristics and requirements a detailed implementation document can be made, which will offer a Jericho enabled network design for a specific company.

Within the logical phase it becomes clear how the functionalities can be implemented in the different suggested Trust broker models. The context that is used to transform the logical components is the difference between in and outside. This context is chosen because of the similarity between the three Trust broker models. Based on the context for these models it becomes evident where each functionality has to be performed and to which non-functional requirements it has to comply. With this information a specific network design can be made, which will play its part in the implementation process. Additionally, the remaining problem of how to secure a loosely coupled set of services is tackled in the logical phase. This is dealt with by suggesting three models in which a secure orchestration is created.

The fourth phase of the IAF model is the physical phase. This phase is further elaboration of the previous book, in which I concluded that the current technologies were not yet mature enough. In this book insight about the maturity level is provided within the technology matrix - which largely consists of the same technologies as suggested before – by adding a readiness level. This readiness level is explained and elaborated in 'appendix D – Technology matrix'.

Winding up this book, the Trust broker has been presented in three 'real life' business scenarios. These scenarios exemplify only a small number of the possibilities the Trust broker will enable in the corporate environment.

A final conclusion to this book can not be given as it doesn't answers a research question. It has however, tackled the research objective: to create a solid foundation for the physical layer. This has been done by answering questions about the four layers of the Integrated Architecture Framework (IAF). This has given new insights in how to implement a concept such as the Trust broker. In addition to that it has raised a few new questions that have to be researched before the realization of a de-perimeterized network is possible.

Future research

The new questions that have arisen during the completion of this book are summarized below.

Future research has to be done in the domains of:

- To which extent can security and auditablity measures go without violating standing privacy regulations?
- How does an infringement on your personal privacy relate to the infringement on your professional identity?
- How can an automated dynamic policy be created that can perform the following tasks:
    - checking and determining which policy rules overrule other policy rules in case of a 'multiple policy check'
    - how policy rules can be made comparable to each other
    - how policy rules can be made on the fly, that are based on the business needs and requirements
- How can a contract be generated automatically?
- Which business incentives have to be used to create a critical mass for adopting a Trust broker and using the trust management system? Because the reputation and importance of the system will depend to a large extent on the number of entities that are using it.
- On a lighter note. Can you call this system a trust broker?

  This question is posed because the system doesn't actually trade in trust. As trust is not transitive (cannot be passed from person to person),not distributive (cannot be shared) and not symmetric (I trust you does not equal you trust me). (*Drs. S. Slone, 2007)*

  The system as proposed now, only creates a context in which a entity can adopt an attitude towards an other entity, which in other words could also be called a trust-level. But to really adopt this vision you have to trust the 'Trust broker'. So one could also name this system a 'trusted broker' or just 'the system that enables me with the appropriate information to create an opinion about an entity', but from a marketing point of view this is not a real fancy name.

Appendix A: Architecture methodology

Extracted from, (Enterprise, Business and IT Architecture and the Integrated Architecture Framework, Capgemini, Andrew L Macaulay, 2006)

Architecture frameworks

- A Holistic View

Clients and the industry as a whole are moving towards a standard (but not yet universally defined and agreed) set of terms that describe different types of architecture. These typically encompass terms such as Enterprise Architecture, Solutions Architecture, and even Security or Governance Architectures as well as the more usual Technical, Applications or Business Architectures.

The following diagram illustrates how Capgemini relates these types of architecture to one another, demonstrating the inclusion of Business Architecture within a full Enterprise Architecture, as well as the need for Solution Architecture to span from Business to Technology.

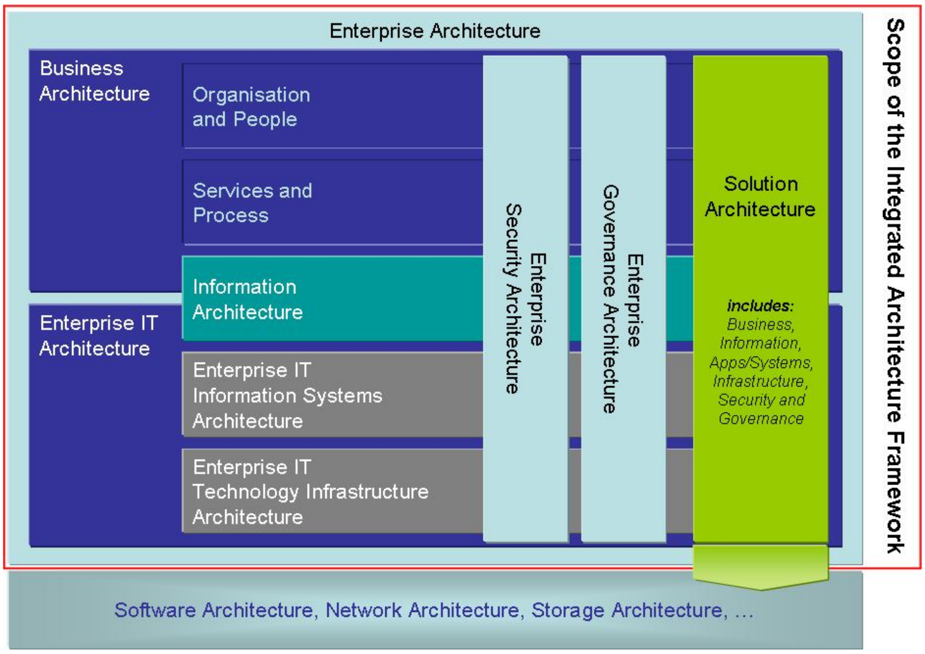

Enterprise Architecture
Business Architecture
Organisation and People
Services and Process
Information Architecture
Enterprise IT Architecture
Enterprise IT Information Systems Architecture
Enterprise IT Technology Infrastructure Architecture
Enterprise Security Architecture
Enterprise Governance Architecture
Solution Architecture
includes: Business, Information, Apps/Systems, Infrastructure, Security and Governance
Scope of the Integrated Architecture Framework
Software Architecture, Network Architecture, Storage Architecture, …

Figure 1. Types of Architecture

It is important to note that each type of architecture will address different levels and types of insight that may span business, information, systems, etc. Within this model Capgemini recognises:

- **Enterprise Architecture** details the structure and relationships of the Enterprise, its business models, the way an organisation will work, and how and in what way Information and ICT will support the organisation's business objectives and goals. Enterprise

Architecture provides an all-encompassing, holistic end-to-end view of the business in terms of people, process, governance and technology within (and external to) the business support those objectives and goals.

- **Enterprise Business Architecture** (or **Business Architecture**) defines the integrated structure of the overall business itself (in terms of organisation, people and process and resources). Business architecture supports business change with a more holistic perspective. This approach is becoming more important with the move towards Service Oriented Architecture at the business level, often termed the Service Oriented Enterprise.
- **Enterprise IT Architecture** defines and describes the structure and relationships of IT systems at the Enterprise level, in terms of the way that IT supports the organisation in achieving its business goals. This typically includes standards and guidelines that are applied within Solution Architectures.
- **Solution Architecture** defines an architecture for a specific solution, whether this be Business or IT. The Solution Architecture provides structure, standards and guidance for the detailed design of a solution and is typically guided by the Enterprise Architecture. Note that "Solution Architecture" is often used as shorthand for "Solution IT Architecture" and is sometimes referred to as Project Architecture.
- **Governance Architecture** defines not only the traditional IT Systems Management capabilities, organisation and systems, but also addresses business governance (how to manage the overall business processes, formal and informal) as well – critical in these days of increasing business regulation and compliance.

Security Architecture defines not only traditional IT security but also addresses business and information security as well as the resulting organisational and business-related services to deliver the required security, often linked to the Governance aspects to cover what is often termed security management.

This holistic view of architecture is directly reflected in Capgemini's approach, the Integrated Architecture Framework (IAF), with specific "Aspect Areas" that focus on Business, Information, Information Systems, Technology Infrastructure, Security and Governance.

IAF, TOGAF and IEEE 1471-2000

During the evolution of IAF, and the various other industry standards, many areas have converged in terms of overall positioning. IAF is compatible with The Open Group Architecture Framework15 (TOGAF) and with IEEE 1471-200016 "Recommended Practice for Architectural Description of Software-Intensive Systems". As an example, the following key definitions are taken directly from TOGAF version 8.1, Developing Architecture Views (Introduction): "The architecture of a system is the system's fundamental organization, embodied in its components, their relationships to each other and to the environment, and

[15] . Information about TOGAF is available from http://www.opengroup.org/togaf

[16] . Information about IEEE 1471-2000 standard can be found on http://www.ieee.org

the principles guiding its design and evolution.” “A view is a representation of a whole system from the perspective of a related set of concerns.” Both of these key definitions are compatible with the use of these terms within IAF, especially when you consider “system” to cover both Business and IT “systems”. It is also worth noting that both of these definitions are based on, and compatible with, those definitions in IEEE 1471-2000. In looking at TOGAF it is clear that it provides best-practice around the organization and processes required for an Enterprise Architecture function within an organization, but does not restrict the architecture models and language to be applied – allowing organizations to select the most appropriate for their business. As IAF itself provides a complete and consistent way to describe the architecture, which is applied for different clients/in different contexts using the relevant Engagement Roadmap, TOGAF can be used as the basis for the process and organization for an architecture function (providing the “Engagement Roadmap” for IAF) with IAF providing the architectural models and language.

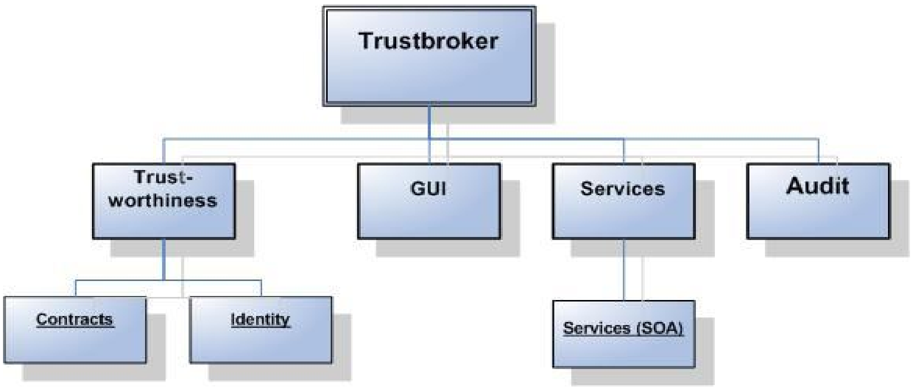
IAF was also formally accepted by the Open Group, as a recognized method within their IT Architect Certification (ITAC) program.

[1]. Information about TOGAF is available from http://www.opengroup.org/togaf

[2]. Details about the Zachman Framework for Enterprise Architecture is available from http://www.zifa.com

Appendix B: The relationship between the Trust broker and SOA

The Trust broker is not a competitor of SOA, instead it is more a addition to the Service Oriented Architecture. The Trust broker adds a level of trust (management) to a SOA environment. This is done by laying more emphasis on identity and contracts, because these two functions will form the foundation of building a trust relationship.



The figure above shows how the Trust broker framework integrates SOA.

The main philosophy behind this Trust broker framework is that it will enable SOA to truly deliver its services across their local boundaries. The Trust broker can achieve this by creating and maintaining a trust relationship with each entity - a user, device or company - it deals with. This trustworthiness is based on two attributes, namely identity and contracts.

The identity attribute specifies the following about an entity; authentication, reputation and behaviour. This acquired information tries to reflect how trust relationships are created in the real world. Based on who you see, what this persons behaviour is and how other people refer to this particular person you are willing to enter some sort of an agreement – how insignificant it may be.

The contract attribute will deliver an agreement between this entity and the services it wants to use from the SOA environment. This agreement can variate from real world contracts till an behaviour code at a forum, anything is possible. The contract attribute will enable a particular level of legal control – based on the agreement – to the company of the Trust broker. Legal control will on one hand enforce entities to obey to the rules set by the agreement or policy of a company and on the other hand will give a company a level of insurance if something goes wrong.

Based on the trustworthiness, a entity can use the SOA environment of any number of companies that is connected to the Trust broker. A key requirement to let a entity use multiple services across multiple SOA environments is Role Based Access. A single entity with different roles will enable a single-sign-on environment across multiple domains, even if the service will only be used once – especially useful in case of intermediary services. Furthermore roles will simplify policy management, this is achieved by increasing the manageability and reducing the administration efforts. Without SOA the Trust broker can't exist, but to use the full potential of an SOA environment across local boundaries a trust management system is necessary – thus a Trust broker.

The connection with SOA and the Trust broker is further explained through the process of Service (Discovery service/ servicebus).

One advantage of SOA is to reuse services for other benefits. However, knowing which services are ready to reuse within a SOA environment remain best practices, which can be concluded according to Rich Rogers, Senior Technical Staff Member at IBM.[17]

During this book I had an idea that could help to define which services could be reused, at least from a technical perspective. My thought was to use the rules of normalization created by Ted Codd18 for relational databases. The goal of database normalization is to minimize redundancy of data, the goal for which it will be used here is to minimize redundancy of services.

The approach will be to define the most basic elements of each functionality and see whether these services can be re-used. The 'third normal form' (TNF) will be enough in order to achieve the goal of minimizing redundancy, because “Every non-prime attribute of the table must be non-transitively dependent on every candidate key.” (19) Or in plain English, this third form ensures that each service only depends upon itself so its relationships with other services can be easily changed.

The three database normalization rules adapted to services are given here:

First Normal Form (FNF)

“A relation is in first normal form if:

Each cell is single-value (there are no repeating groups or arrays)

Entries in a column (field) are or of same kind.

There are no duplicate rows in the table

The requirement that there be no duplicated rows in the table means that the table has a key (although the key might be made up of more than one column-even, possibly, of all the columns).” (from ayzec.com) For my service normalization I will interpreted the FNF as;

There are no repeating services or functionalities, unless it is required.

Second Normal Form (SNF)

“A relation is in second normal form if it is in first normal form and it has no partial dependencies. Partial dependencies?

Partial Dependency: A partial dependency may occur in a relation only if it contains two or more key fields (a composite primary key). If a non-key field is dependant on only part of the composite key and not all the key fields, this is called a partial dependency.” (from ayzec.com)

For my service normalization I will interpreted the SNF as;

No service is dependent from an other services or functionality, unless it is part of the in or output of the service.

Third Normal Form (TNF)

“A relation is in third normal form if it is in second normal form and if it has no transitive dependencies.

[17] http://www.ibm.com/developerworks/webservices/library/ws-reuse-soa.html accessed on September 2007

[18] http://en.wikipedia.org/wiki/Ted_Codd accessed on September 2007

[19] Codd, E.F. "Further Normalization of the Data Base Relational Model."

Transitive Dependency: A transitive dependency can only occur in a relation where there is more than one non-key field. If there is a dependency among non-key fields, this is called a transitive dependency." (from ayzec.com). For my service normalization I will interpreted the TNF as;
No service is transitive dependent from an other services or functionality.

Administered to the Trust broker Framework

The services – as given below – are rough and must be refined if these were to be implemented. However in reality each of these services is further specified by more smaller services, this depends on soft- and hardware choices. Because these smaller services are very technical based it will not be split up, further in this book.
In order to refine these services I will try to use the method that is used when normalizing a database. To goal of database normalization is to minimize redundancy of data, the goal for which it will be used here is to minimize redundancy of services.
Given are the following services:

Trustworthiness,
Identity, authentication, federation, user-centric,
Reputation, skills, references, ext. sources, past behaviour,
Behaviour, current behaviour, scans, oblige to contract/ agreement scan, non-authorized access scan, Control, fulfillment-scan, generating contracts/ agreements, policy, updates at ext. sources or predefined circumstances, contracts --> trusted timestamps --> digital signature,

***IT Governance*, ** complying to regulations, management dashboard, monitoring, generating reports, control, reporting, audit,

Policy, policy based on role, encryption, end-point security, authorization, policy combined with data-classification, multiple policies checked,

SOA, SOSA, sandboxing, meta-model, mashup, orchestration, services, enterprise servicebus,

***Service Discovery*,** discover policies from multiple companies, SOA
determining best service --> policy --> service,

Security, multi level security, classification levels, data sensitivity

Unfortunately I failed in applying these rules for normalization to the services I had already defined. This due to the fact that the services are formulated on a high level, and are more aimed at the business/ organizational process then towards the technology. Because when I tried, the services that remained were the same ones as in the hierarchal scheme, so this wasn't really helpful.

In conclusion, the rules of database normalization by Ted Codd are not particularly useful for defining which services can be reused. The main reason for this lays in the nature of services, they are designed to for one or more entities. Because of this the service is often made to specific for one user group. With this the rules of normalization can also be useful, they can help to identify the smallest unique service.

Although the idea of creating rules to rule out duplicates and combining similar services remains – in my humble opinion – a good idea.

Managing services in an dynamic environment

When services are implemented in your company there must be a system that can manage these services. In time these services will be deployed within another context to fulfill new tasks. But how can all these services migrate quickly to their new context? To accomplish this it must be known which dependencies a services has with other services, business and technical ones a like.

A possible solution for this problem can be found in the software world. Linux is very advanced in using a package management system[20], two of the most successful or advanced are apt-get/ aptitude[21] and conary[22]. *"In such a system, software is distributed in packages, usually encapsulated into a single file. As well as the software itself, packages often include other important information, such as the full name, a description of its purpose, the version number, vendor of the software,* checksum *information, and a list of other packages, known as* dependencies*, that are required for the software to run properly."*[23]

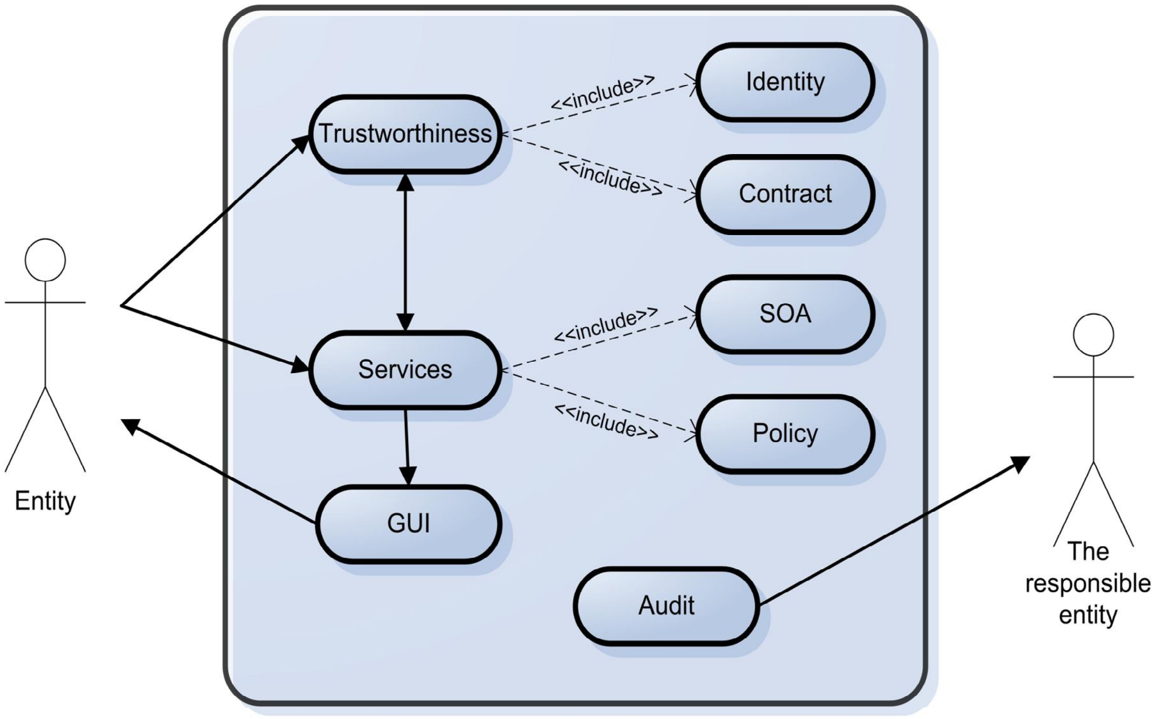
This data could easily be adapted to information relevant for services within a SOA environment, where the information about dependencies is probably the most valuable source of information. Information about this can have two advantages;

- One, it offers and maintains a clear view of how services are used within – and for the future outside – a company.
- Two, it creates a way to easily reuse services, because it becomes very clear and surveyable to all actively involved parties.

[20] http://en.wikipedia.org/wiki/Package_management_system accessed September 2007
[21] http://en.wikipedia.org/wiki/Apt-get accessed September 2007
[22] http://wiki.rpath.com/wiki/Conary accessed September 2007
[23] http://en.wikipedia.org/wiki/Package_management_system accessed September 2007

Appendix C: Diagrams

Use case diagram



Use case name:	Trustworthiness
Goal:	Determines the trust level of an entity.
Pre-condition:	Non
Post-condition:	Has established a Trust Context Report (with substance)
Description:	• Need to establish a Trust Context Report (trust level) with the use of a few Security Attributes (parameters) so it can share this information with Services. • Establishing trust is based on identity and contracts (but is extensible) • Trust Context Report (trust level) is defined by all modules under Trustworthiness.
Exception:	Non, it will always deliver a Trust Context Report.

Use case name:	Services
Goal:	To deliver the best fitting service to an entity.
Pre-condition:	Entity has an trust level
Post-condition:	It has supplied a service which for filled to all policy, security and requirements of the entity.

Description:	• Will supply all the data and services that are offered by the company or CoT. • Is extensible by adding new modules • Gives/ channels its information back to the GUI
Exception:	Service is not started/supplied when trust level is to low.

Use case name:	GUI
Goal:	To display or channel the information between the entity and the system.
Pre-condition:	Has to know the role and service it has to present.
Post-condition:	Non
Description:	• Displays results, data or services. • Displays this information in a highly customizable way (to the need of a entity, and maybe based on a role, trust level or policy).
Exception:	Will not present new information when policy/ security is broken.

Use case name:	Audit
Goal:	To give the responsible entities the necessary information.
Pre-condition:	A way to collect all this information is required.
Post-condition:	Non
Description:	• Displays the results from the all the audit modules that are implemented in the network. It can be compared with a management dashboard. • To accomplish this all kinds of monitoring and control tools are used. This information will be checked on whether certain requirements are met.
Exception:	Non

Use case name:	Identity
Goal:	To establish a trust relation with the entity.
Pre-condition:	Must be willing to share information
Post-condition:	Output is done in several security attributes.
Description:	• Determines, based on the modules beneath it, the trustworthiness of an entity and will forward this information to Trustworthiness. o Trustworthiness of an entity consists basically of: ▪ The identity, thus authentication. ▪ The Reputation about the identity. • Output is an accumulation of security attributes
Exception:	Non (that I can think of)

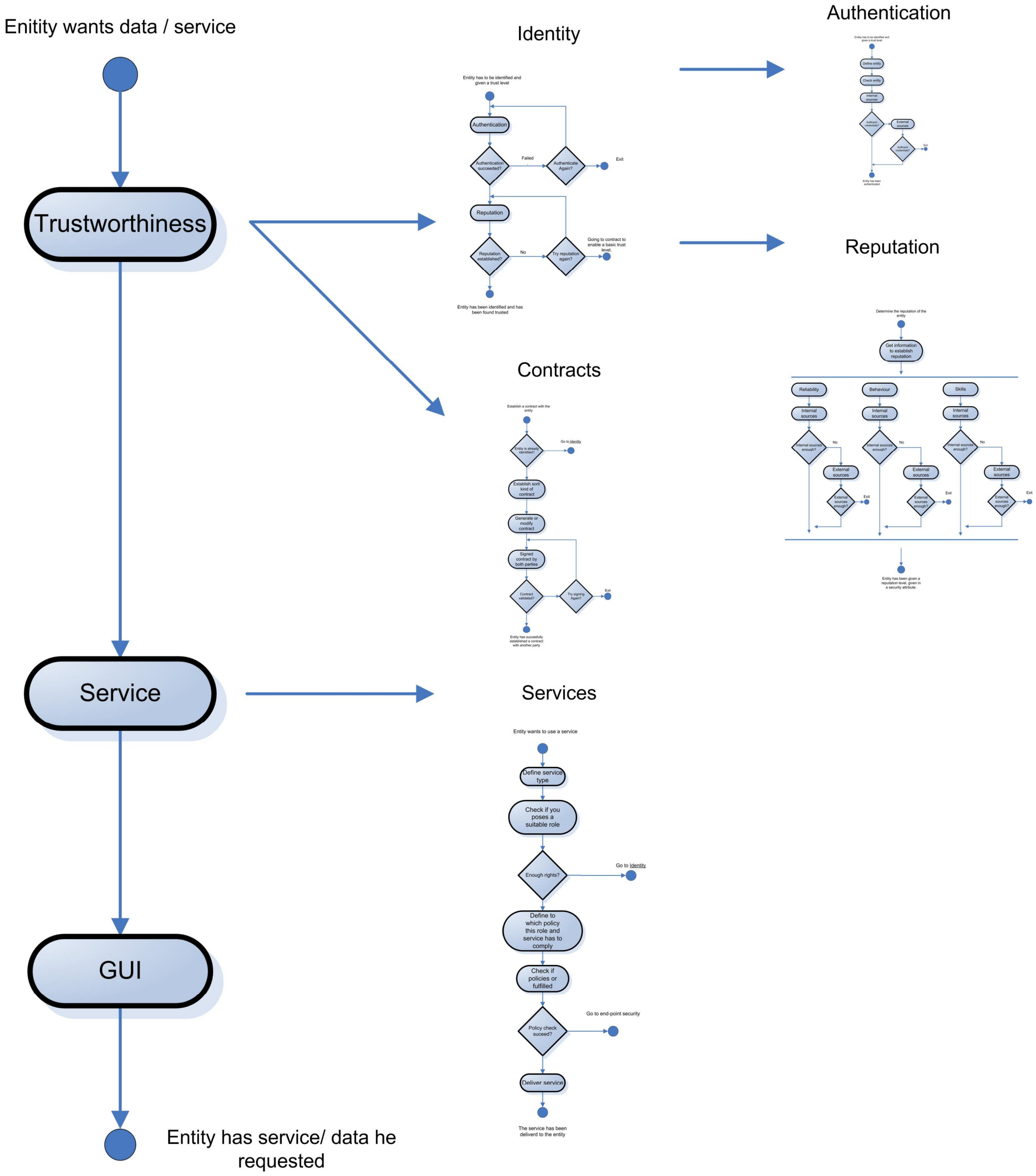
Use case name:	Contracts
Goal:	To establish an agreement between two or several parties, which gives them some enforceability.
Pre-condition:	Entity is identified

Post-condition:	Contract or agreement is formulated in a way it can be monitored.
Description:	• Determined by the business needs & requirements of all relating companies. o For very important issues – i.e. a partnership between two companies - this process has to be done manually (as long as computers cannot compare different sorts of information within a certain context) • If a contract is made it will store information about: o The expectations the other party has from you o The obligations the other party has towards the other party(ies) o It will monitor the behaviour of the different parties and will compare this with the expectations & obligations. (may require some human input) • Output is an accumulation of security attributes
Exception:	Entity – or the role it facilitates – does not has the clearance to sign the contract.

Use case name:	SOA (discovery service & servicebus)
Goal:	An architecture that is able to deliver any service, based upon some requirements that are given by at least two several parties.
Pre-condition:	Entity has been identified and found trustworthy.
Post-condition:	Results are channeled to the GUI.
Description:	• Within SOA a discovery service determines which services will best serve the needs of an entity, based on the information it has about this entity and the role in which it is acting. o This information will be received from Trustworthiness by means of a Trust Context Report, such as the limitations of a contract (which obligations), which identity (what skills and reputation), etc. o Information from trustworthiness can be used to determine the role off the entity. • Based on the role the corresponding policy levels will be applied. • Within SOA the Service bus will make it possible to integrate any service - through the use of open standards - in the Trust broker framework.
Exception:	Entity does not for fill the obligations of a policy.

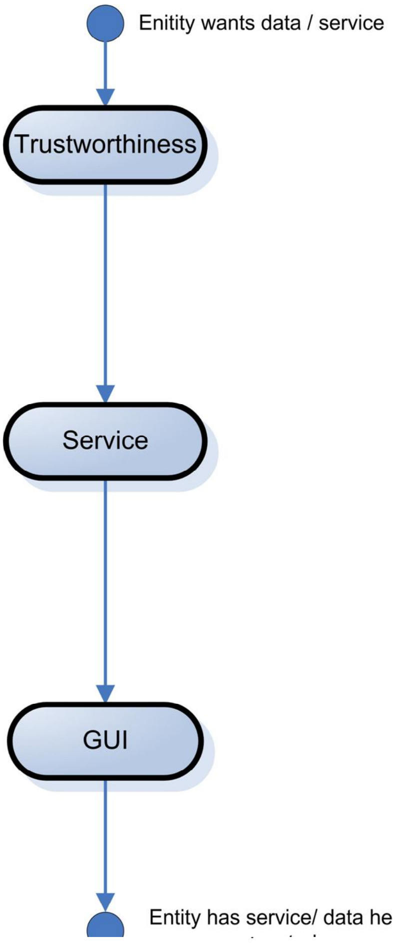
Use case name:	Policy
Goal:	To bring together and decide which policy in which role and circumstance has to be used.
Pre-condition:	Entity has been identified, found trustworthy and has selected a service(type).
Post-condition:	Keeps scanning after service delivery to create reputation and auditing information.
Description:	• The Policy module is an assembly from all the different policies that several companies or applications demand. o Policy uses the role, in which an identity acts, that is determined in the Discovery Service or established earlier in the process, i.e. with the authentication at a company. Based on this role it enables the corresponding authorization process.
Exception:	Entity does not come through the policy check.

The main model overview


Enitity wants data / service
Trustworthiness
Service
GUI
Entity has service/ data he requested
Identity
Enitity has to be identified and given a trust level
Authentication
Authentication succeeded?
Failed
Authenticate Again?
Exit
Reputation
Reputation established?
No
Try reputation again?
Going to contract to enable a basic trust level.
Entity has been identified and has been found trusted
Authentication
Enitity has to be identified and given a trust level
Define entity
Check entity
Internal sources
Sufficient credentials?
External sources
Sufficient credentials?
Exit
Entity has been authenticated
Reputation
Determine the reputation of the entity
Get information to establish reputation
Reliability
Behaviour
Skills
Internal sources
Internal sources enough?
No
External sources
External sources enough?
Exit
Entity has been given a reputation level, given in a security attribute.
Contracts
Establish a contract with the entity
Entity is already identified?
Go to Identity
Establish sort/ kind of contract
Generate or modify contract
Signed contract by both parties
Contract validated?
Try signing Again?
Exit
Entity has succesfully established a contract with another party
Services
Entity wants to use a service
Define service type
Check if you poses a suitable role
Enough rights?
Go to Identity
Define to which policy this role and service has to comply
Check if policies or fulfilled
Policy check suceed?
Go to end-point security
Deliver service
The service has been deliverd to the entity

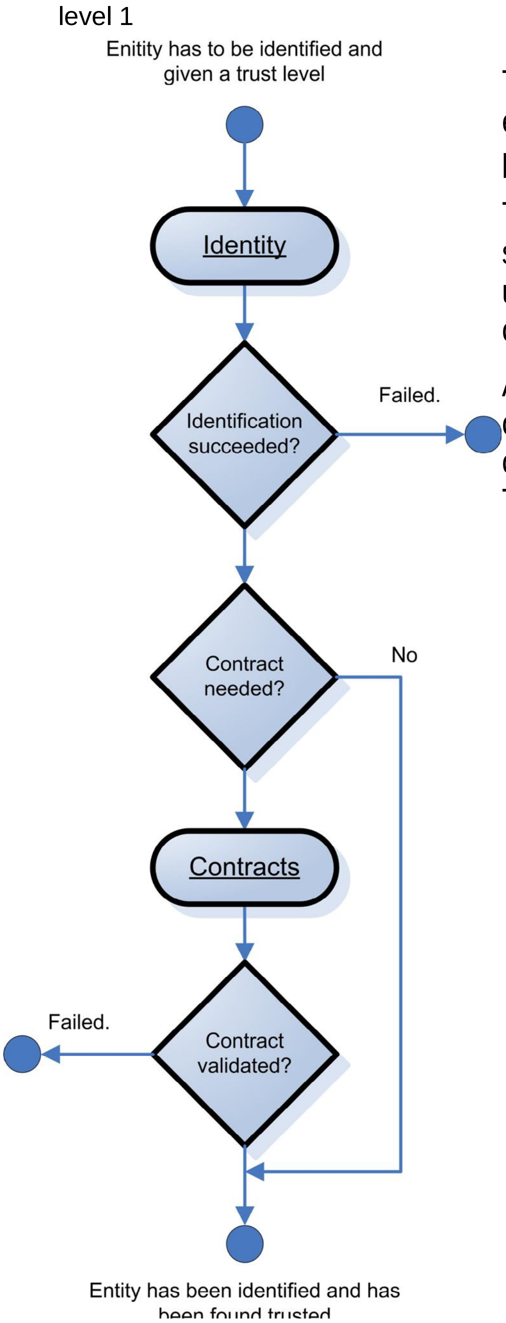
This illustration gives an overview of the activity diagrams that are presented below. As is shown with trustworthiness, the main activity diagram is further specified in two lower levels.

Main model



This main model shows a very high level overview of the Trust broker framework. This model shows that to get a service you first have to be found trustworthy for the system, then the most suitable service of your demands is chosen – if you fulfill to all requirements – and finally this information is presented to the GUI service.

Trustworthiness


level 1
Enitity has to be identified and given a trust level
Identity
Identification succeeded?
Failed.
Contract needed?
No
Contracts
Contract validated?
Failed.
Entity has been identified and has been found trusted

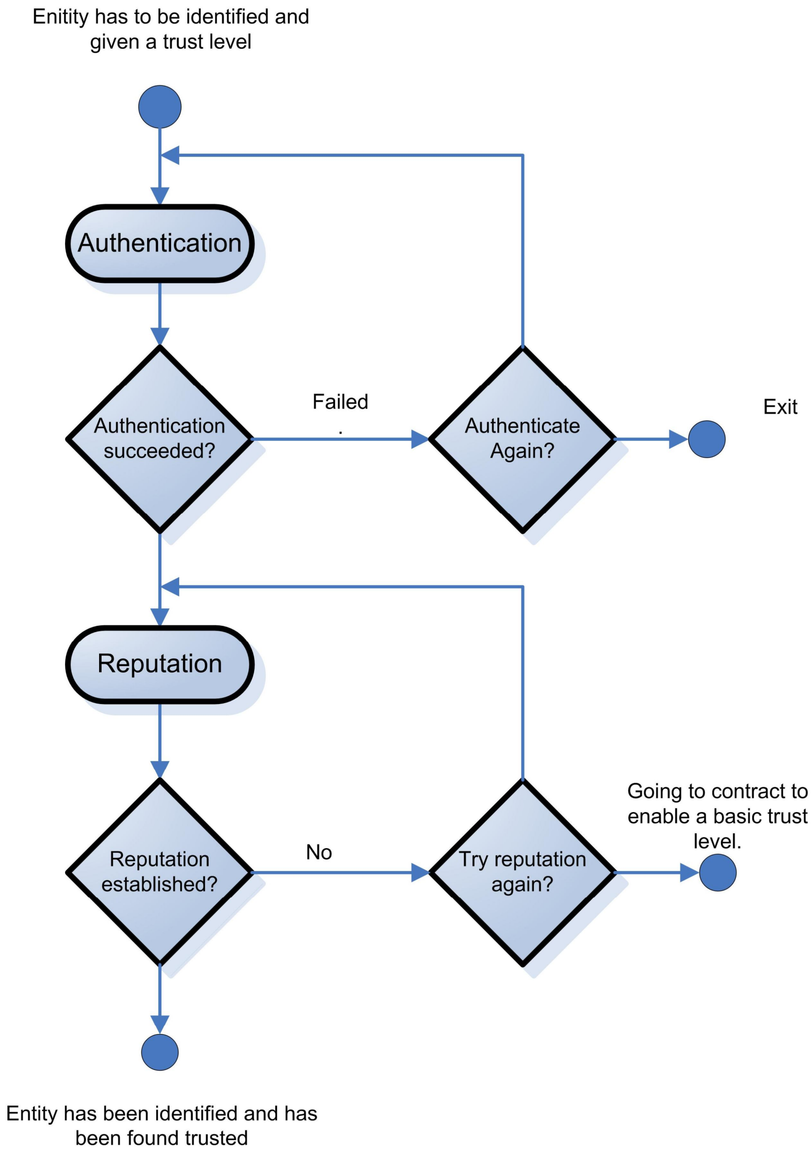
Trustworthiness main goal is to determine the trust level of an entity. This process, as shown on the left, is done by two processes.

The process of Identity is explained further on, but the outcome should always be successful. However when for unfamiliar/ unknown reasons the Identity process is not successful it is also covered on a higher level.

After this a selection is made whether or not the identity has to create a contract for a higher trust level. When necessary the contract process is started, otherwise the process of Trustworthiness is ended and goes back to the main level.

Identity

level 2


Enitity has to be identified and given a trust level
Authentication
Authentication succeeded?
Failed
.
Authenticate Again?
Exit
Reputation
Reputation established?
No
Try reputation again?
Going to contract to enable a basic trust level.
Entity has been identified and has been found trusted

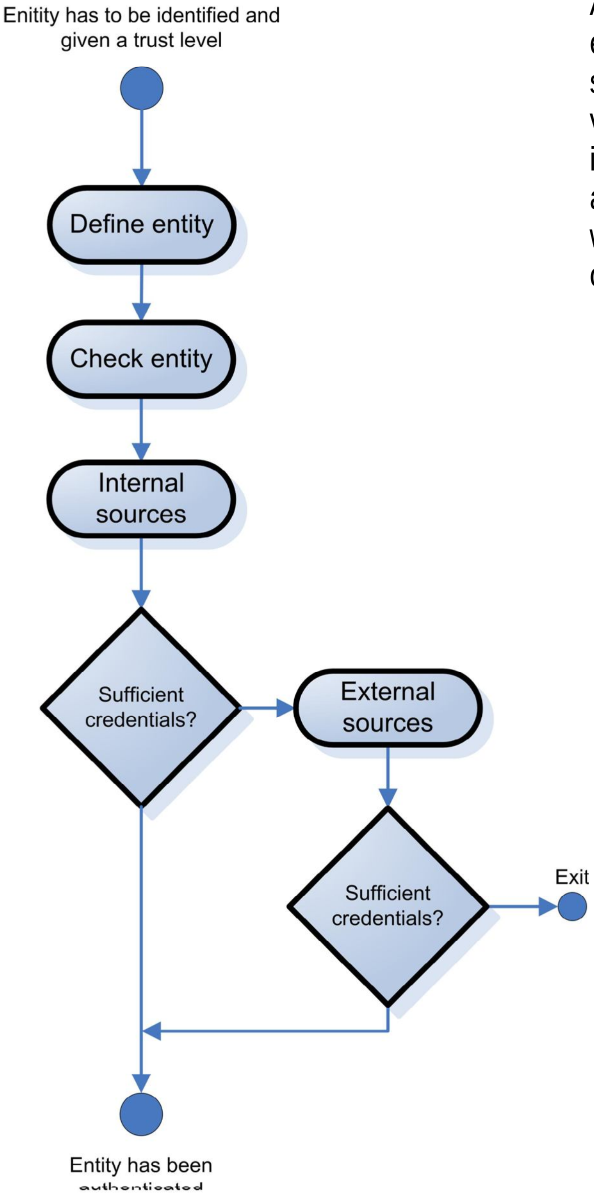
The goal of the Identity process is to establish a trust relationship with an entity. This is done by two processes, namely Authentication and Reputation.

With Authentication a check for anomalies is done whether the Authentication process was really succeeded. If not the entity has the choice to retry or exit the process.

After the entity is authenticated a reputation check will be done. This process is also checked on two levels whether or not the reputation was established. And if not the entity is given the same choice as above, retry or not.

Authentication

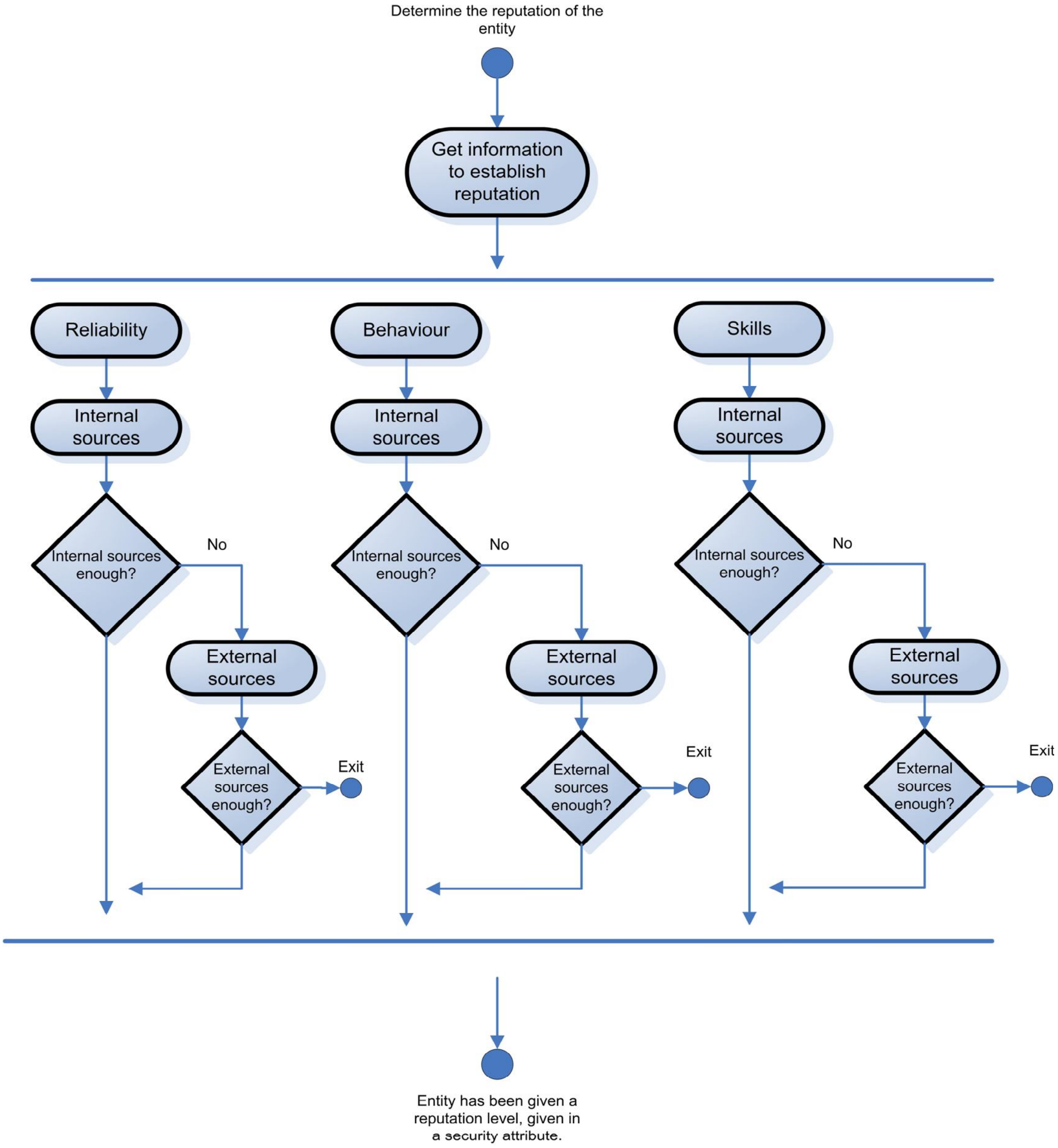
level 3



Authentication has the goal to determine whether the entity is who it says it is. This is similar with the current solutions. However the way an entity authenticates can vary within this model, it is even possible that the information needed to authenticate an entity is not available at the local Trust broker. When this happens it will check the external sources – these are often sources of partners – if they can authenticate the entity.

Reputation

level 3

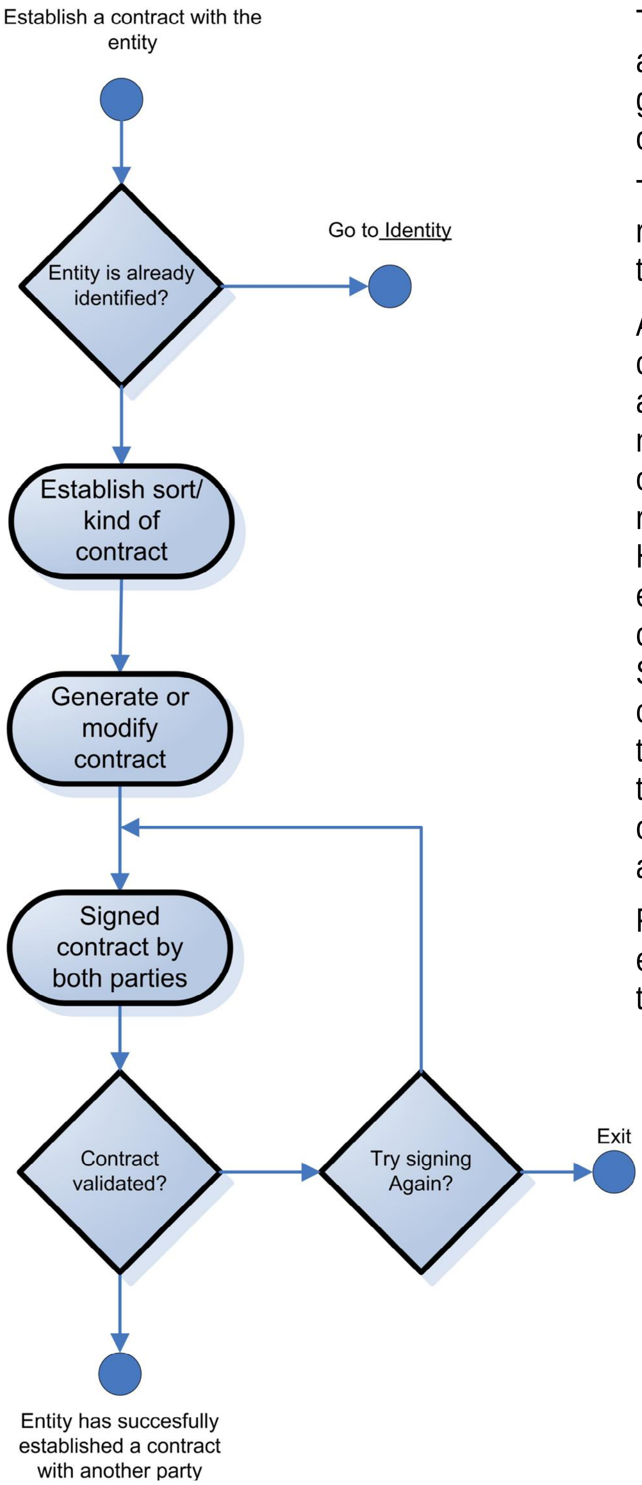

Determine the reputation of the entity
Get information to establish reputation
Reliability
Behaviour
Skills
Internal sources
Internal sources enough?
No
External sources
External sources enough?
Exit
Entity has been given a reputation level, given in a security attribute.

The main goal of the Reputation process is to give Trustworthiness and supply the necessary security attributes so it can establish a trust level. Hence, the goal of Reputation is really about collecting information about the authenticated entity. I can imagine that only information about the professional role of this entity is needed.

The process that is started within will collect all information that is needed. In order to enable Trustworthiness, Reputation has to collect information about the reliability, behaviour and skills of the authenticated entity. If this is not sufficient enough, information from external sources and partners of the Trust broker are checked.

Contract

level 2


Establish a contract with the entity
Entity is already identified?
Go to Identity
Establish sort/ kind of contract
Generate or modify contract
Signed contract by both parties
Contract validated?
Try signing Again?
Exit
Entity has succesfully established a contract with another party

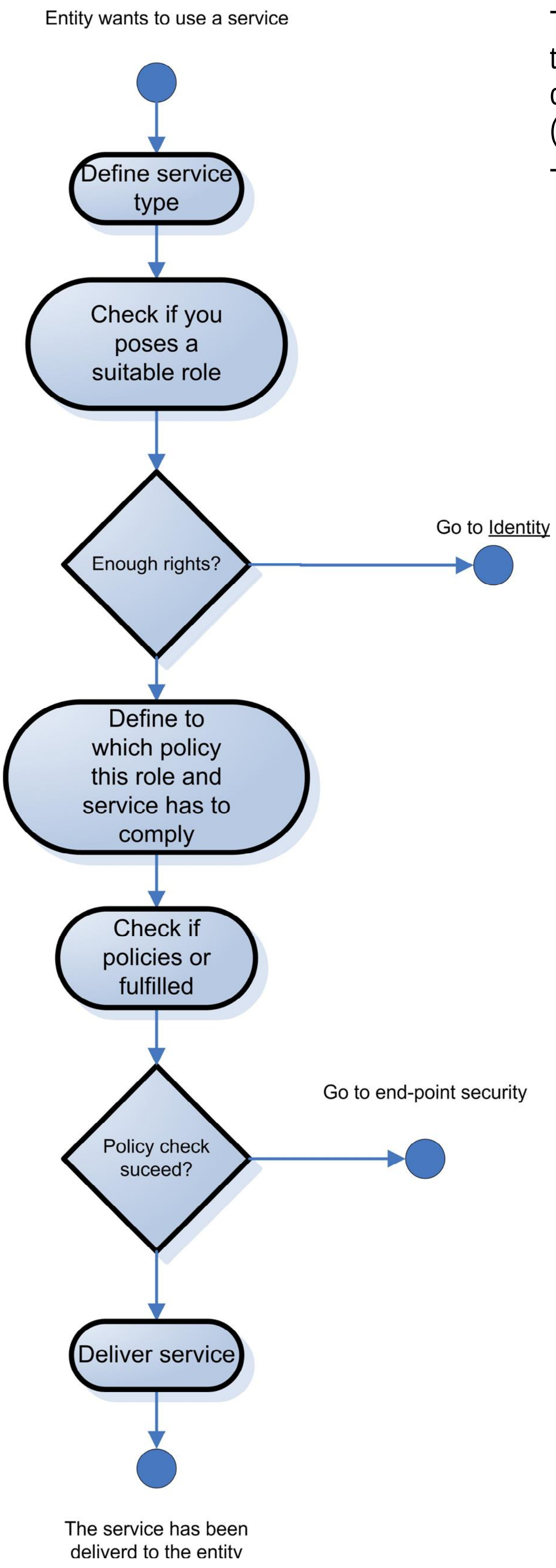
The goal of the Contract process is to establish an agreement between two or several parties, which gives them both some enforceability over the other one(s).

The process is started with a check if the entity is really identified, this means authenticated and given a trust level.

After this a process is started to identify which kind of contract is needed. To do this, a (small) risk assessment is necessary to determine the impact the new relationship may have. If this is high risk an official contract is established. If the impact is relatively low it can be done with an agreement. However, it does not matter which kind of contract is established, the most important part is that this contract is signed by a legally binding signature. Secondly, which is almost equally important, the contract must be given a trusted time stamp. Such a time stamp records the date of creation in such a way that the time can not be altered, not even by the owner him self. This is very useful in administrative and legal processes.

Furthermore, the processes are checked whether everything was done correctly, if not the entity gets the opportunity to retry.

Services


Entity wants to use a service
Define service type
Check if you poses a suitable role
Enough rights?
Go to Identity
Define to which policy this role and service has to comply
Check if policies or fulfilled
Policy check suceed?
Go to end-point security
Deliver service
The service has been deliverd to the entitv

level 1

The Services process has the goal to deliver any service to any authenticated user - with a certain role – that complies to all the policies within the network of partners (CoT).

This process is done by:

1. Defining (asking the entity) which kind of service is requested/ demanded,
2. Checking which role – that the authenticated entity has – is suitable to start that service. This check is more about comparing clearance levels.
3. Checking if the role really has enough rights.
4. Defining to which policy the role has to comply. So far the Trust broker knows the service type and the role which wants to use the service.
5. Checking whether the role complies with the policy that comes together with the requested service.
6. If the policy check fails the entity is referred to another process, for example end-point security.
7. If all processes are successful, the service is delivered (to the next process, GUI).

Appendix D: Technology matrix for the Trust broker Framework

This technology matrix is based on the technologies as suggested in *'Jericho in depth... Trust broker services'* chapter (10), however during time some technologies where added. Although the Trust broker uses many services, not all belong to the Trust broker framework itself, therefore I will only discuss the most necessary functionalities and their corresponding technologies. In the conclusion of *'Jericho in depth... Trust broker services'* I said that there are enough technologies to create an initial Trust broker, but not all technologies were mature enough. Within this technology matrix I will include a readiness level that is based on ten demands.

Ten demands to determine technology readiness level

Question	Purpose
1. Who are the designers and/or the manufacturers? ● It is a big and respectable company, (10pt) ● It is a steady manufacturer, (5pt) ● It is a newcomer.(1 pt)	This is to determine if the technology has some future. Although smaller companies can have a silver bullet, the chance that a big and respectable company continues to support a product is higher. Conclusion, it is about reliability of the product (and not in a technical way).
2. Is there a large community within the industry? ● There is an active world based community, (10 pt) ● There is an local community, (5 pt) ● There is small community, (1 pt)	The community is largely connected to the adoption of the industry, but community has another meaning. A community is more about a group of people that continue to support the technology. In case of open source – and in some extent open standards – this can be particularly useful. For example, a company already bankrupt or has abandoned the project, but the community will keep supporting it.
3. Is there a large adoption of this technology within the industry? ● There is a large adoption within industry, (10 pt) ● Within a select group it is widely adopted, (5 pt) ● There is virtually no adoption, (0 pt)	A company can be the biggest in the market and its technology can protect us from the end of the universe, but if nobody uses the product it means nothing. So, how many people are using this technology and how does its future looks like.
4. Is it based on open standards? ● It is completely based on open standards, (10 pt) ● It supports open standards, (5 pt) ● It only uses proprietary standards, (0 pt)	Comply to Jericho Forum Commandment number four.
5. Is it compatible and inter-operable with other technologies?	Although open standards improve the chance of interoperability it does not necessarily mean it can inter-operate with different technologies. So with this question

● It is compatible with certain other technologies, (5 pt) ● It can not operate with other technologies, (0 pt)	I assume the out-of-the-box capability of this technology.
6. Are there alternatives to prevent a vendor lock-in? ● There are more then three serious alternatives, (10 pt) ● There are more then two alternatives, (5 pt) ● There are no real alternatives for this technology, (0 pt)	To prevent a vendor lock-in and stimulate innovation and quality of the technology there must be some alternatives of this product.
7. Does it have(long-term) support? ● It has long-term (minimal 5 years) support, (10 pt) ● It has support, (5 pt) ● It has no explicit support, (0 pt)	In order to use it in a professional business environment the technology must have support, preferably long-term support.
8. Is it (well) documented? ● It is documented extensively, (10 pt) ● It is documented, (5 pt) ● It is hardly or poorly documented, (0 pt)	Next to good support from the manufacturer of the technology it is equally important that the product has good documentation.
9. Is the company supporting a loosely-coupled set of services? ● The company is already producing in a service oriented manner, (10 pt) ● The company is migrating or thinking about service orientation, (5 pt) ● There is no development in service orientation, (0 pt)	A product does not necessarily has to be developed as a service, but it illustrates that the company is innovating and adapting to a new market demands.
10. How important is security in this technology? ● It is designed to be secure, (10 pt) ● It can be secured if you want, (5 pt) ● It is difficult to secure (properly), (0 pt)	How is the product or technology developed? Only with a certain functionality in mind and security added afterwards, or is security build into the technology.

Each question has been divided in three sub-questions that will determine a rough level in which the product satisfies to the main question. Each of this sub-questions are coupled with a number, in total a technology can get a score of one hundred (100), where 100 is the best score a product can work.

All questions are worked out for each technology

Readiness level of the technologies

The first row displays the technology and on the left the numbers of the questions.

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	tot
OpenID	5	10	5	10	10	5	0	5	10	10	65
MS Cardspace	10	1	0	5	5	5	10	10	10	5	60
Higgins	10	5	0	10	10	5	0	5	10	10	65
XRI	5	5	0	10	10	5	0	10	10	0	55
SAML 2.0	10	5	10	10	10	5	10	10	10	10	90
Jyte	0	1	0	10	10	10	0	0	10	0	41
Ebay	10	5	0	0	0	10	10	0	5	0	40
Experian	5	0	10	0	0	5	0	0	10	10	40
Trustplus	1	0	0	5	10	5	0	5	5	5	36
Link contracts	10	5	0	10	10	5	5	10	10	10	75
WS-Agreement	10	5	0	5	10	0	5	10	10	10	65
WS-Policy	10	10	10	5	10	0	10	10	10	10	75
XACML	10	10	10	10	10	0	10	10	10	10	90
WSDL	10	10	10	5	10	5	10	10	10	10	90
WS-Choreography	10	10	5	5	10	0	10	10	10	10	80
WS-Discovery	10	0	5	5	10	0	10	10	10	10	70
WS- Trust	10	10	10	5	10	0	10	10	10	10	85
WS-Security	10	10	10	5	10	0	10	10	10	10	85
WS-Federation	10	10	10	5	10	5	10	10	10	10	90
ID-WSF	10	10	5	10	10	5	10	10	10	10	90
ID-FF	10	5	5	10	10	5	10	10	10	10	90
SOAP	10	10	10	10	10	5	10	10	10	10	90

These numbers don't represent how good a technology is only its readiness grade for adoption in the market. Each of the above mentioned technologies has the potential to be used to create the Trust broker Framework functionality.

Results

Primary Function	Functionality	Technology	Readiness grade
Trust-worthiness	Identity	OpenID	65
		MS Cardspace	60
		Higgins	65
		XRI	55
		SAML 2.0	90
	Reputation	Jyte	41
		Ebay Experian	40 40
		Trustplus	36
	Behaviour	Lot of information can be gathered from monitoring tools from the Audit or End-point security functionalities.	
	Contracts	Link contracts (XDI)	75
		WS-Agreement	65
Services	Policy	WS-Policy	75
		XACML	90
	SOA/ servicebus	WSDL	90
		WS-Choreography	80
	Discovery service	WS-discovery	70
General		WS-Trust	85
		WS-Security (WSS)	85
		WS-Federation	90
		ID-WSF	90
		ID-FF SOAP	90 90

Elaboration per technology

This elaboration will be based on the available internet information about each specific technology. When appropriate text of the site will be copied. When this is the case links to these sites will be displayed under each header.

OpenID

http://wiki.openID.net/
http://en.wikipedia.org/wiki/OpenID

1. **Who are the designers and/or the manufacturers?**
   OpenID was originally developed by Brad Fitzpatrick of LiveJournal, but the term now also includes the Light-Weight Identity, Yadis, Sxip DIX protocol that was proposed at IETF, and XRI/i-names. Future OpenID specifications are being developed in a meritocratic fashion on openID.net, involving many technology companies, user companies and open-source developers.

   The OpenID Foundation is currently being formed to help manage intellectual property, marketing efforts and other activities related to the success of the OpenID community. The singular goal of the OpenID Foundation is to protect OpenID so that it may be used by any and all that want to.

   A full description of the foundation, its structure and its goals will be forthcoming once the details have been fleshed out. In the mean time, you can learn more from the links below.

   Foundation Board
   The OpenID Foundation is currently being lead by a small board consisting of seven members of the OpenID community. On the board page you can find the minutes of the board meetings.
   Committees
   The board has spun off several committees to work on various tasks.

   OpenID Europe is a non-profitable organization that aims at promoting the open source framework "**OpenID**" in Europe, by participating actively to its development and by facilitating its adoption by the European users and European services (profitable and non profitable).

   Its role is also to play a role in the identity market growth.

2. **Is there a large community within the industry?**
   OpenID is increasingly gaining adoption among large sites, with organizations like AOL acting as a provider. In addition, integrated OpenID support has been made a high priority in Firefox 3[1] and Microsoft is working on implementing OpenID 2.0 in Windows Vista.[2]

3. **Is there a large adoption of this technology within the industry?**
   As of July 2007, there are over 120 million OpenID's on the Internet (see below) and approximately 4,500 sites have integrated OpenID consumer support.[4]

America Online *provides (brokers) OpenIDs, in the form "openID.aol.com/screename".*
idproxy.net *provides OpenIDs for* Yahoo! *users via Yahoo!'s authentication API (BBAuth); idproxy.net was created by a former Yahoo! developer but is not otherwise related to the company.*
Orange *offers OpenIDs to their 40 million broadband subscribers.*
Six Apart *blogging hosts* LiveJournal *and* Vox*. Both support OpenID;* Vox *as a provider and* LiveJournal *as both a provider and a relying party.*
Other services accepting OpenID as an alternative to registration include Wikitravel*, photo sharing host* Zooomr*, linkmarking host* Ma.gnolia*, identity aggregator* ClaimID*, icon provider* IconBuffet*,* Basecamp *and* Highrise *by* 37signals*, and* Jyte*.*
OpenID Enabled *lists many more sites, services, and platforms.*

4. **Is it based on open standards?**
   The official site currently states:
   Nobody should own this. Nobody's planning on making any money from this. The goal is to release every part of this under the most liberal licenses possible, so there's no money or licensing or registering required to play. It benefits the community as a whole if something like this exists, and we're all a part of the community.

5. **Is it compatible and inter-operable with other technologies?**
   Sun Identity Provider for OpenID https://openID.sun.com/opensso/index.jsp
   OpenID + CardSpace = Ubiquity
   It's great to see Bill Gates announce that CardSpace will integrate with OpenID. Convergence, and momentum, is building. http://journals.aol.com/panzerjohn/abstractioneer/entries/2007/02/08/openID--cardspace--ubiquity/1403

6. **Are there alternatives to prevent a vendor lock-in?**
   Vendor lock-in can't happen because openID is supported by a group of vendors.
   Although each system is different in approach the following technologies could be seen as alternatives SAML2.0 & cardspace.

7. **Does it have(long-term) support?**
   Support is given by the foundation and the Europe foundation, problems are dealt with by them or the community.

8. **Is it (well) documented?**
   As it is open source its documentation is good, but scattered. All supporting vendors have some support on the subject, and within the community additional questions can be resolved.

9. **Is the company supporting a loosely-coupled set of services?**
   OpenID is not especially designed for it, but supports it very well. This is because it is focused on web-services and has an decentral design.

10. **How important is security in this technology?**
    The technology is not entirely safe yet, but a lot of improvements are made in this field.
    http://www.disruptivetelephony.com/2007/03/is_openID_reall.html

MS Cardspace
http://en.wikipedia.org/wiki/Windows_CardSpace
http://cardspace.netfx3.com/content/faq.aspx#54

1. **Who are the designers and/or the manufacturers?**
   Microsoft
2. **Is there a large community within the industry?**
   Not yet, but has the potential as it is shipped with windows 2003, vista and .net framework 3. In addition it is also available for windows XP.
3. **Is there a large adoption of this technology within the industry?**
   Many other vendors support or want to be interoperable with this technology.
4. **Is it based on open standards?**
   Is CardSpace Microsoft proprietary?
   CardSpace is part of Microsoft's implementation of an identity metasystem supported by open standard WS-* protocols. While CardSpace runs on Microsoft Windows, it is compliant with the supported WS-* standards and with other vendors' implementations on other platforms. In addition, other vendors can build implementations of CardSpace-like technologies to run on other platforms.
   http://cardspace.netfx3.com/content/faq.aspx#54
5. **Is it compatible and inter-operable with other technologies?**
   Because it is token-agnostic, CardSpace does not compete directly with other Internet identity architectures like OpenID and Liberty Alliance. In some ways the three approaches to identity can be seen as complementary. [1]
   In February 2006, IBM and Novell announced that they will support the Higgins trust framework to provide a development framework that subsumes a support for the Web Services Protocol Stack underlying CardSpace within a broader, extensible support for other identity-related technologies, such as SAML and OpenID.
6. **Are there alternatives to prevent a vendor lock-in?**
   As mentioned in number five it will be supported and extended in the Higgins framework, so this could be seen as an alternative.
7. **Does it have(long-term) support?**
   Because the technology has Microsoft behind is back it will surely be supported for the next five years.
8. **Is it (well) documented?**
   One of the major advantages of Microsoft is that it documents everything in its MSDN library. So yes it is documented very well.
9. **Is the company supporting a loosely-coupled set of services?**
   How does CardSpace relate to SOA?
   CardSpace is an essential component of any application which needs to identify the user, regardless of any given architectural perspective.

10. **How important is security in this technology?**
    Does CardSpace provide end-to-end security features? In other words, with CardSpace, will users have to employ any other security mechanisms to share personal information securely? E.g security tokens, two factor authentication, etc.?
    CardSpace is a useful substrate that provides much of what is needed to implement a secure identity infrastructure. Additional enhancements, such as n-factor authentication, biometrics, etc, *are openly welcome and can be seamlessly integrated into the metasystem.*
    How does CardSpace interoperate with existing security protocols?
    CardSpace is part of Microsoft's implementation of an identity metasystem conforming to Kim Cameron's widely accepted "Identity Metasystem". The metasystem is fully supported by the WS-* security protocols and is open to all parties. CardSpace uses these protocols to provide a secure way in which the release of identity claims can be controlled by a user and trusted by a receiving application or service.

Higgins
http://en.wikipedia.org/wiki/Higgins_project
http://swik.net/higgins
http://www-03.ibm.com/press/us/en/pressrelease/19280.wss

1. **Who are the designers and/or the manufacturers?**
   The initial code for the Higgins Project was written by Paul Trevithick in the summer of 2003. In 2004 the effort became part of SocialPhysics.org, a collaboration between Paul and Mary Ruddy, of Parity Communications, Inc., and John Clippinger, a senior fellow at the Berkman Center of the Harvard Law School. Higgins, under the original name **Eclipse trust framework**, was accepted into the Eclipse Foundation in early 2005. Mary and Paul are the project co-leads. IBM and Novell's participation in the project was announced in early 2006. Higgins has received technology contributions from IBM and Novell as well as from several other firms and individuals. The project plans a 1.0 release at the end of summer 2007.
2. **Is there a large community within the industry?**
   Because the technology is supported by IBM, novell and other smaller companies it has a potentially large community. In addition it is managed by another large open source community on the internet, namely Eclipse.
3. **Is there a large adoption of this technology within the industry?**
   Nothing is published on this fact, but IBM and Novell are working on implementing this technology in their own identity manager products.
4. **Is it based on open standards?**
   It is entirely open source.
5. **Is it compatible and inter-operable with other technologies?**
   It is very inter-operable! This is the first user-centric identity management effort to follow the open source software model, where hundreds of thousands of developers contribute -- and continually drive improvements through collaborative innovation. Being an open source effort, Higgins will support any computer running Linux, Windows or any operating system,

and will support any identity management system.

6. **Are there alternatives to prevent a vendor lock-in?**
The alternative is cardspace, but as it is open source and supported by at least four companies a vendor lock-in is not very likely.

7. **Does it have(long-term) support?**
Although nothing is official many companies support it indirect (by letting some employees work full time on this product). The open source community of Eclipse however will support it as long as is has momentum.

8. **Is it (well) documented?**
Very well, only a bit scattered as they use several technologies.

9. **Is the company supporting a loosely-coupled set of services?**
Higgins is a framework that will enable users and enterprises to integrate identity, profile, and relationship information across multiple systems. Using service adapters, existing and new systems such as directories, collaboration spaces, and communications technologies (e.g. Microsoft/IBM WS-*, LDAP, email, IM, etc.) can be plugged into the Higgins framework. Applications written to the Higgins API can virtually integrate the identity, profile, and relationship information across these heterogeneous systems. A design goal is that Higgins be useful in the development of applications accessed through browsers and rich clients. Our intent is to define Higgins in terms of service descriptions, messages and port types consistent with an SOA model and to develop a Java binding and implementation as an initial reference.

10. **How important is security in this technology?**
Security is a main issue from the beginning. Higgins uses several security measures in its design. Important exemple's are ws-security, saml, etc.

XRI
http://en.wikipedia.org/wiki/XRI

1. **Who are the designers and/or the manufacturers?**
developed by the XRI Technical Committee at OASIS.

2. **Is there a large community within the industry?**
Not a very big community, but there are companies trying to use this technology.
http://www.inames.net/

3. **Is there a large adoption of this technology within the industry?**
No, not yet. But many think it is a very good technology.

4. **Is it based on open standards?**
Yes, as it is part of the OASIS group.

5. **Is it compatible and inter-operable with other technologies?**
Yes, because it is based on the URI technology it can be used for all kinds of other uses, thus interoperable.

6. **Are there alternatives to prevent a vendor lock-in?**
   There are no direct alternatives, but it is open source.
7. **Does it have(long-term) support?**
   Depends on the companies supporting it.
8. **Is it (well) documented?**
   It is pretty extensive documented by the OASIS group.
   http://wiki.oasis-open.org/xri
9. **Is the company supporting a loosely-coupled set of services?**
   The protocol is based on the idea of URI's, so yes it is based on services.
10. **How important is security in this technology?**
    Security is not standard with URI technology, so not directly implemented in XRI. However this document,http://www.oasis-open.org/committees/download.php/2523/xri-requirements-and-glossary-v1.0.doc, shows that they are busy with implementing this.

SAML2.0

1. **Who are the designers and/or the manufacturers?**
   OASIS approved version 2.0 of the Security Assertion Markup Language (SAML) as a standard, providing guidelines for developers to create single sign-on applications that work across disparate locations on the Internet.
   Backed by vendors, such as IBM, BEA Systems and Sun Microsystems, SAML 2.0 lets users authenticate data exchanges between an application and a security system, paving the way for the exchange of Web services (define). Web services allow applications to communicate with each other regardless of boundaries on the Web. http://www.internetnews.com/dev-news/article.php/3489786
   After version one it was further developed by OASIS, Liberty and Shibbolet.
2. **Is there a large community within the industry?**
   This is hard to find out, no obvious communities around saml 2.0 exists only corporate communities, but this is the adoption rate.
3. **Is there a large adoption of this technology within the industry?**
   Yes, the following companies are using saml in one way or another:
   Baltimore Technologies, BEA Systems, Computer Associates, Entrust, Hewlett-Packard, Netegrity, Oblix, OpenNetwork, Reactivity, RSA Security, SAP, Sun Microsystem and Google apps.
   http://xml.coverpages.org/saml.html
4. **Is it based on open standards?**
   As saml is an standard of OASIS and the mission of OASIS is,
   “OASIS (Organization for the Advancement of Structured Information Standards) is a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society.” SAML is an open standard.

5. **Is it compatible and inter-operable with other technologies?**
   SAML has become the definitive standard underlying many web Single Sign-On solutions in the enterprise identity management problem space.
   SAML assumes the *principal* (often a user) has enrolled with at least one identity provider. This identity provider is expected to provide local authentication services to the principal. However, **SAML does not specify the implementation of these local services; indeed, SAML does not care how local authentication services are implemented** (although individual service providers most certainly will).
   Thus a service provider relies on the identity provider to identify the principal. At the principal's request, the identity provider passes a SAML assertion to the service provider. On the basis of this assertion, the service provider makes an access control decision.
   http://en.wikipedia.org/wiki/SAML
   So yes it is inter-operable.

6. **Are there alternatives to prevent a vendor lock-in?**
   There are no real alternatives, actually non. However since it is an open standard this isn't an issue.

7. **Does it have(long-term) support?**
   This fact isn't documented somewhere, but since major vendors are using this technology they contribute to the development of it. Furthermore Liberty, OASIS and all supporting vendors continue to improve the open standard.

8. **Is it (well) documented?**
   It is pretty well documented. For example the papers of OASIS about SAML 2.0
   Security Assertion Markup Language (SAML) v2.0
   The complete SAML v2.0 OASIS Standard set (PDF format) and schema files are available.
   The approved specification set consists of:

   - Assertions and Protocols
   - Bindings
   - Profiles
   - Metadata
   - Authentication Context
   - Conformance Requirements
   - Security and Privacy Considerations
   - Glossary

9. **Is the company supporting a loosely-coupled set of services?**
   Since the protocol doesn't care to which authentication protocols it has to talk it is independent to other services, so yes it supports the philosophy of SOA. However without other technologies it won't deliver the expected function, so it is still dependent on other services.

10. **How important is security in this technology?**
    As described in the documentation it is designed to be secure in dealing with sensitive assertions. So yes, security is important.

Jyte

1. **Who are the designers and/or the manufacturers?\**
   Who made Jyte?
   The company is called JanRain. JanRain's other main product is MyOpenID. See also this claim.

   JanRain is a team of 12 Internet ninjas who are building the next generation of light-weight identity services for the web. We have been leading the way in developing the emerging OpenID protocol (one username, one password for all of the sites you go to on the Internet) as a means of delivering those services. JanRain's mission is to provide identity services that help users securely manage, control and innovate their own digital identities with OpenID.
2. **Is there a large community within the industry?**
   Exact numbers aren't know, but it has grown quickly since it was only introduced in 31 jan. 2007.
3. **Is there a large adoption of this technology within the industry?**
   Since it isn't a business critical services it isn't supported in business at all.
4. **Is it based on open standards?**
   It isn't truly open source or standard, but it is free since you can use it's api to perform your own tasks.
5. **Is it compatible and inter-operable with other technologies?**
   Since it is based on a webservice this is pretty well.
6. **Are there alternatives to prevent a vendor lock-in?**
   Yes, some of them are discussed below.
7. **Does it have(long-term) support?**
   Since it is dependent of its developer this is not entirely sure.
8. **Is it (well) documented?**
   NO
9. **Is the company supporting a loosely-coupled set of services?**
   Yes. It is based on web technology and has a separate api.
10. **How important is security in this technology?**
    Not very important. However since it depends on openID one form of security is fixed.

eBay

1. **Who are the designers and/or the manufacturers?**
   Ebay
2. **Is there a large community within the industry?**
   No, just eBay.

3. **Is there a large adoption of this technology within the industry?**
   Again, only eBay.
4. **Is it based on open standards?**
   NO
5. **Is it compatible and inter-operable with other technologies?**
   Nothing, know about this.
6. **Are there alternatives to prevent a vendor lock-in?**
   Yes, such as Jyte, Trustplus, Experian, etc
7. **Does it have(long-term) support?**
   Since it is one of the main functionalities that differentiates eBay from the other market places this is probably a yes.
8. **Is it (well) documented?**
   NO
9. **Is the company supporting a loosely-coupled set of services?**
   Ebay is not delivering api's, but it is based on web technology so in theory they can work with other services.
10. **How important is security in this technology?**
    Not very, although it is not in the media. Security of the reputation system is not a primary business need of of eBay, but to deliver transactions/ auction.

Experian

1. **Who are the designers and/or the manufacturers?**
   By the company experian it self. Although it isn't really an application, it is an example of how an aggregation of information about credit and marketing information on consumers and businesses can create a context on which an entity can make a decision.
2. **Is there a large community within the industry?**
   There is no real community that stimulates experian in developing their functionality. Unless you define a community in a degree of users, but the user side is descriped in the adoption grade.
3. **Is there a large adoption of this technology within the industry?**
   Experian has a really large client base;
   “Experian supports clients in over 65 countries and across a broad range of industry sectors. Some of our client relationships have been in place for over 25 years. “
4. **Is it based on open standards?**
   No
5. **Is it compatible and inter-operable with other technologies?**
   Not known, they only deliver the information. How experian deals with this information is not exactly publicly available.
6. **Are there alternatives to prevent a vendor lock-in?**

There are alternatives, but not with the same accuracy.

7. **Does it have(long-term) support?**
   They don't have to give support as they only supply information.
8. **Is it (well) documented?**
   No.
9. **Is the company supporting a loosely-coupled set of services?**
   Yes, you can do anything with this information.
10. **How important is security in this technology?**
    Very, as their main product is information it is very important to secure this information.

Trustplus

1. **Who are the designers and/or the manufacturers?**
   Trustplus itself
2. **Is there a large community within the industry?**
   Not known
3. **Is there a large adoption of this technology within the industry?**
   Not known
4. **Is it based on open standards?**
   It is free and has its open api, which standards it uses is not exactly known but it is obviously largely based on web technologies.
5. **Is it compatible and inter-operable with other technologies?**
   Through their api it is.
6. **Are there alternatives to prevent a vendor lock-in?**
   As it is a new market not many vendors are offering such technologies, and those who are present their functions in different ways.
7. **Does it have(long-term) support?**
   Not publicly available.
8. **Is it (well) documented?**
   At the moment not really.
9. **Is the company supporting a loosely-coupled set of services?**
   Yes, this is partially done through their api.
10. **How important is security in this technology?**
    Although they say it is security isn't high on the list and is therefor only marginally supported.

Linking contracts (XDI)

http://www.oasis-open.org/committees/xdi/faq.php

1. **Who are the designers and/or the manufacturers?**
   OASIS is the main manufacturer, but the two main developers **are** XDI TC organizers Drummond Reed, Cordance and Geoffrey Strongin, AMD.
2. **Is there a large community within the industry?**
   OASIS it self has a large group of supports within the corporate and individual context, but there is no specific large community around XDI.
3. **Is there a large adoption of this technology within the industry?**
   No, not yet since it is still in development.
4. **Is it based on open standards?**
   Yes totally, this is one of the main characteristics of OASIS.
5. **Is it compatible and inter-operable with other technologies?**
   Yes, it is very interoperable with other web technologies, like SOAP, etc.
6. **Are there alternatives to prevent a vendor lock-in?**
   No alternatives today, but it is an open standard.
7. **Does it have(long-term) support?**
   Nothing kwown about it, but since it has an big corporation behind it (OASIS) it will be supported for the moment. This support means that it will keep being developed, not that they will send people to you in order to install and configure it for your company.
8. **Is it (well) documented?**
   Yes, it is very well documented, see the OASIS site about XDI as given above.
9. **Is the company supporting a loosely-coupled set of services?**
   Since it is open standard and compatible with other technologies it supports SOA, as a matter of fact it is especially build for it.
10. **How important is security in this technology?**
    Very! They even support linking contracts and a specially designed Privacy Framework. But all other kinds of typical security issues are covered.

WS-Agreement

http://searchsoa.techtarget.com/originalContent/0,289142,sid26_gci1042406,00.html

http://forge.ogf.org/sf/projects/graap-wg

1. **Who are the designers and/or the manufacturers?**
   The WS-* specs are specifications that IBM works on with partners such as BEA, Oracle, Microsoft and others.
2. **Is there a large community within the industry?**
   Since the introduction of a developers toolkit from IBM the community has grown in the area of ws-*. Although ws-agreement is a very new protocol, final version in may 2007, there are already some implementations known (as is listed at the second site).
3. **Is there a large adoption of this technology within the industry?**

No not yet, but again it is very new technology.

4. **Is it based on open standards?**
Yes and no, it is open but has a very special license. This is because Microsoft is a partner. Besides the license the end goal is the same, it is open.
5. **Is it compatible and inter-operable with other technologies?**
It is made to be compatible. And it is based on XML.
6. **Are there alternatives to prevent a vendor lock-in?**
No.
7. **Does it have(long-term) support?**
Because it is a consortium of companies this can be expected, but it depends on its success in the market.
8. **Is it (well) documented?**
Very well, each party has its own documentation on it.
9. **Is the company supporting a loosely-coupled set of services?**

   Yes since it is based on XML.
10. **How important is security in this technology?**
Very, it is designed to interact and cooperate with all kinds of ws-* security protocols.

WS-Policy

http://www.ibm.com/developerworks/library/specification/ws-polfram/

http://www.w3.org/Submission/WS-Policy/

http://en.wikipedia.org/wiki/WS-Policy

1. **Who are the designers and/or the manufacturers?**
Contributors: IBM, BEA Systems, Microsoft, SAP AG, Sonic Software, VeriSign
2. **Is there a large community within the industry?**
There is, if I analyze all kinds of documents, it has a potentially large developers base. But I cannot say this with certainty.
3. **Is there a large adoption of this technology within the industry?**
According to http://www.w3.org/2002/ws/policy/ there are already some adoptions of ws-agreement.
4. **Is it based on open standards?**
Yes, just like all ws-* technologies.
5. **Is it compatible and inter-operable with other technologies?**
Yes, designed to be.
6. **Are there alternatives to prevent a vendor lock-in?**
No, not to this extend.

7. **Does it have(long-term) support?**
Because it is a consortium of companies this can be expected, but it depends on its success in the market.
8. **Is it (well) documented?**
Yes, as can be seen on every partners site, for example msdn.
9. **Is the company supporting a loosely-coupled set of services?**
Yes, is based on XML.
10. **How important is security in this technology?**
Very, it is designed to interact and cooperate with all kinds of ws-* security protocols.

XACML

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

http://en.wikipedia.org/wiki/XACML

http://xml.coverpages.org/xacml.html

1. **Who are the designers and/or the manufacturers?**
BEA Systems, Booz Allen Hamilton, Computer Associates, Entrust, Gluecode Software, IBM, Sun Microsystems, and Others Advance Open Standard for Information Access Control.
2. **Is there a large community within the industry?**
Yes, many support the technology.
3. **Is there a large adoption of this technology within the industry?**
Yes, many companies use it or are developing it even further.
4. **Is it based on open standards?**
Yes, it is registered at OASIS.
5. **Is it compatible and inter-operable with other technologies?**
Designed to be.
6. **Are there alternatives to prevent a vendor lock-in?**
No.
7. **Does it have(long-term) support?**
Many supports, so yes.
8. **Is it (well) documented?**
Yes
9. **Is the company supporting a loosely-coupled set of services?**
Yes (XML)
10. **How important is security in this technology?**
It is designed to deliver it, so very important!

WSDL

http://www.w3.org/TR/wsdl

http://en.wikipedia.org/wiki/Web_Services_Description_Language

1. **Who are the designers and/or the manufacturers?**
   Contributors: IBM, BEA Systems, Microsoft, SAP AG, Sonic Software, VeriSign
2. **Is there a large community within the industry?**
   Yes. It is becoming a de facto.
3. **Is there a large adoption of this technology within the industry?**
   Many companies use the technology already.
   http://api.google.com/GoogleSearch.wsdl
4. **Is it based on open standards?**
   Yes, just like all ws-* technologies.
5. **Is it compatible and inter-operable with other technologies?**
   Yes, designed to be.
6. **Are there alternatives to prevent a vendor lock-in?**
   Yes, but are becoming out of date.
7. **Does it have(long-term) support?**
   Because it is a consortium of companies this can be expected, but it depends on its success in the market.
8. **Is it (well) documented?**
   Yes, as can be seen on every partners site, for example msdn.
9. **Is the company supporting a loosely-coupled set of services?**
   Yes, is based on XML.
10. **How important is security in this technology?**
    Very, it is designed to interact and cooperate with all kinds of ws-* security protocols.

WS-Choreography

http://www.w3.org/TR/ws-chor-model/

http://www.service-architecture.com/web-services/articles/ws_choreography_description_language_cdl.html

http://www.looselycoupled.com/glossary/WS-Choreography

http://en.wikipedia.org/wiki/Web_Service_Choreography

1. **Who are the designers and/or the manufacturers?**
   Contributors: IBM, BEA Systems, Microsoft,

BPML, now BPMN

BPSS by ebXML[3]
WSFL by IBM
XLANG by Microsoft
BPEL4WS by IBM, Microsoft and BEA

2. **Is there a large community within the industry?**
   Yes
3. **Is there a large adoption of this technology within the industry?**
   Some major vendor are incorporating it in their products, like SAP.
4. **Is it based on open standards?**
   Yes, just like all ws-* technologies.
5. **Is it compatible and inter-operable with other technologies?**
   Yes, designed to be.
6. **Are there alternatives to prevent a vendor lock-in?**
   No.
7. **Does it have(long-term) support?**
   Because it is a consortium of companies this can be expected, but it depends on its success in the market.
8. **Is it (well) documented?**
   Yes, as can be seen on every partners site, for example msdn.
9. **Is the company supporting a loosely-coupled set of services?**
   Yes, is based on XML.
10. **How important is security in this technology?**
    Very, it is designed to interact and cooperate with all kinds of ws-* security protocols.

WS-Discovery

http://en.wikipedia.org/wiki/Web_Services_Dynamic_Discovery

http://schemas.xmlsoap.org/ws/2005/04/discovery/

http://xml.coverpages.org/ni2004-10-29-a.html

1. **Who are the designers and/or the manufacturers?**
   The WS-Discovery standard was developed by BEA Systems, Canon, Intel, Microsoft, and WebMethods.
2. **Is there a large community within the industry?**
   Not yet, but it is growing through the promotion of the developing companies.
3. **Is there a large adoption of this technology within the industry?**
   All developing companies are using it in their new product, if it is suitable ofcourse.
4. **Is it based on open standards?**

Yes, just like all ws-* technologies.

5. **Is it compatible and inter-operable with other technologies?**
   Yes, designed to be.
6. **Are there alternatives to prevent a vendor lock-in?**
   No
7. **Does it have(long-term) support?**
   Because it is a consortium of companies this can be expected, but it depends on its success in the market.
8. **Is it (well) documented?**
   Yes, as can be seen on every partners site, for example msdn.
9. **Is the company supporting a loosely-coupled set of services?**
   Yes, is based on XML.
10. **How important is security in this technology?**
    Very, it is designed to interact and cooperate with all kinds of ws-* security protocols.

WS-Trust

http://www.ibm.com/developerworks/library/specification/ws-trust/

http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html

1. **Who are the designers and/or the manufacturers?**
   Contributors: IBM, BEA Systems, Microsoft, Layer 7 Technologies, Oblix, VeriSign, Actional, Computer Associates, OpenNetwork Technologies, Ping Identity, Reactivity, RSA Security
2. **Is there a large community within the industry?**
   There is.
3. **Is there a large adoption of this technology within the industry?**
   Major companies are using it in their networks, since it is becoming a standard for security with web services.
4. **Is it based on open standards?**
   Yes, just like all ws-* technologies.
5. **Is it compatible and inter-operable with other technologies?**
   Yes, designed to be.
6. **Are there alternatives to prevent a vendor lock-in?**
   No, not really.
7. **Does it have(long-term) support?**
   Because it is a consortium of companies this can be expected, but it depends on its success in the market.
8. **Is it (well) documented?**

Yes, as can be seen on every partners site, for example msdn.

9. **Is the company supporting a loosely-coupled set of services?**
   Yes, is based on XML.
10. **How important is security in this technology?**
    Very, it is designed to interact and cooperate with all kinds of ws-* security protocols.

WS-Security (wss)

http://en.wikipedia.org/wiki/Web_Services_Security

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

http://www.ibm.com/developerworks/library/specification/ws-secure/

1. **Who are the designers and/or the manufacturers?**
   Originally developed by IBM, Microsoft, VeriSign and Forum Systems, the protocol is now officially called WSS and developed via committee in Oasis-Open.
2. **Is there a large community within the industry?**
   A very big, it is becoming the de facto for the industry.
3. **Is there a large adoption of this technology within the industry?**
   Yes, if the biggest party in the field of web services technology, the Liberty Alliance has incorporated WSS.
4. **Is it based on open standards?**
   Yes, just like all ws-* technologies.
5. **Is it compatible and inter-operable with other technologies?**
   Yes, designed to be.
6. **Are there alternatives to prevent a vendor lock-in?**
   No, not to this extend.
7. **Does it have(long-term) support?**
   Because it is a consortium of companies this can be expected, but it depends on its success in the market.
8. **Is it (well) documented?**
   Yes, as can be seen on every partners site, for example msdn.
9. **Is the company supporting a loosely-coupled set of services?**
   Yes, is based on XML.
10. **How important is security in this technology?**
    Designed to deliver security!

WS-Federation

http://www.ibm.com/developerworks/library/specification/ws-fed/

http://msdn2.microsoft.com/en-us/library/bb498017.aspx

1. **Who are the designers and/or the manufacturers?**
   Contributors: BEA Systems, BMC Software, CA, Inc., IBM, Layer 7 Technologies, Microsoft, Novell, VeriSign
2. **Is there a large community within the industry?**
   Yes, see sites above.
3. **Is there a large adoption of this technology within the industry?**
   Yes, see the sites above.
4. **Is it based on open standards?**
   Yes, just like all ws-* technologies.
5. **Is it compatible and inter-operable with other technologies?**
   Yes, designed to be.
6. **Are there alternatives to prevent a vendor lock-in?**
   Yes, ID-FF
7. **Does it have(long-term) support?**
   Because it is a consortium of companies this can be expected, but it depends on its success in the market.
8. **Is it (well) documented?**
   Yes, as can be seen on every partners site, for example msdn.
9. **Is the company supporting a loosely-coupled set of services?**
   Yes, is based on XML.
10. **How important is security in this technology?**
    Very, it is designed to interact and cooperate with all kinds of ws-* security protocols.

ID-WSF

http://xml.coverpages.org/ni2005-02-11-b.html

http://developers.sun.com/identity/reference/techart/id-enabled-ws.html

http://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_wsf_2_0_specifications_including_errata_v1_0_updates

1. **Who are the designers and/or the manufacturers?**
   Sun Microsystems
2. **Is there a large community within the industry?**

Yes.

3. **Is there a large adoption of this technology within the industry?**
   There are many companies supporting the ID-WSF specifications.
4. **Is it based on open standards?**
   Yes, it is completely open/
5. **Is it compatible and inter-operable with other technologies?**
   Yes
6. **Are there alternatives to prevent a vendor lock-in?**
   Yes, the ws-* services.
7. **Does it have(long-term) support?**
   The same as with the ws-* services. The Liberty Alliance is a consortium of over 170 companies, so the changes of long-term support are very high.
8. **Is it (well) documented?**
   Yes, see link of liberty above.
9. **Is the company supporting a loosely-coupled set of services?**
   Yes.
10. **How important is security in this technology?**
    Just as WSS it is designed to provide security.

ID-FF

http://vegdave.wordpress.com/category/technology/project-liberty/id-ff/

http://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_ff_1_2_specifications

1. **Who are the designers and/or the manufacturers?**
   Sun Microsystems
2. **Is there a large community within the industry?**
   No really, just a few big ones.
3. **Is there a large adoption of this technology within the industry?**
   Not really since it's functionality is basically the same as SAML 2.0.
4. **Is it based on open standards?**
   Yes, it is completely open.
5. **Is it compatible and inter-operable with other technologies?**
   Yes
6. **Are there alternatives to prevent a vendor lock-in?**
   Yes, the ws-federation and SAML 2.0 services.
7. **Does it have(long-term) support?**
   The same as with the ws-* services. The Liberty Alliance is a consortium of over 170

companies, so the changes of long-term support are very high.

8. **Is it (well) documented?**
   Yes, see link of liberty above.
9. **Is the company supporting a loosely-coupled set of services?**
   Yes.
10. **How important is security in this technology?**
    It is designed to be protected by ID-WSF, so yes it is important.

SOAP

http://en.wikipedia.org/wiki/SOAP

http://www.w3.org/TR/soap/

http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

1. **Who are the designers and/or the manufacturers?**
   Don Box, DevelopMentor
   David Ehnebuske, IBM
   Gopal Kakivaya, Microsoft
   Andrew Layman, Microsoft
   Noah Mendelsohn, Lotus Development Corp.
   Henrik Frystyk Nielsen, Microsoft
   Satish Thatte, Microsoft
   Dave Winer, UserLand Software, Inc.
2. **Is there a large community within the industry?**
   Very big, it is the de facto on the internet.
3. **Is there a large adoption of this technology within the industry?**
   Yes, many companies use it. For example, Google.
4. **Is it based on open standards?**
   Yes.
5. **Is it compatible and inter-operable with other technologies?**
   Yes
6. **Are there alternatives to prevent a vendor lock-in?**
   Yes, but is only for one segment, the servers, Sun's J2EE Blueprints.
7. **Does it have(long-term) support?**
   Because it is a consortium of companies this can be expected, but it depends on its success in the market.
8. **Is it (well) documented?**
   Yes, as can be seen on every partners site, for example msdn.
9. **Is the company supporting a loosely-coupled set of services?**
   Yes, is based on XML.
10. **How important is security in this technology?**

Very, it is designed to interact and cooperate with all kinds of ws-* security protocols.