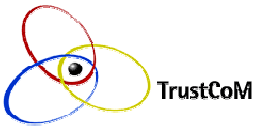

Deliverable

18

VO Trust, Security & Contract Management Framework V1

WP04 Trustcom Framework

Edited by Michael Wilson, CCLRC

July 2005

Draft V4

TrustCoM

A trust and Contract Management framework enabling secure collaborative business processing in on-demand created, self-managed, scalable, and highly dynamic Virtual Organisations

SIXTH FRAMEWORK PROGRAMME

PRIORITY IST-2002-2.3.1.9

Prepared for the European Commission under FP6 Contract No. 01945 as a deliverable form

LEGAL NOTICE

The following organisations are members of the Trustcom Consortium:

Atos Origin,
Council of the Central Laboratory of the Research Councils,
BAE Systems,
British Telecommunications PLC,
Universitaet Stuttgart,
SAP AktienGesellschaft Systeme Anwendungen Produkte in der Datenverarbeitung,
Swedish Institute of Computer Scence AB,
Europaeisches Microsoft Innovations Center GMBH,
Eidgenoessische Technische Hoschschule Zuerich,
Imperial College of Science Technology and Medicine,
King's College London,
Universitetet I Oslo,
Stiftelsen for industriell og Teknisk Forskning ved Norges Tekniske Hoegskole,
Universita degli studi di Milano,
The University of Salford,
International Business Machines Belgium SA .

© Copyright 2005 Atos Origin on behalf of the Trustcom Consortium (membership defined above).

Neither the Trustcom Consortium, any member organisation nor any person acting on behalf of those organisations is responsible for the use that might be made of the following information.

The views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect the views of the European Commission or the member organisations of the Trustcom Consortium.

All information provided in this document is provided 'as-is' with all faults without warranty of any kind, either expressed or implied. This publication is for general guidance only. All reasonable care and skill has been used in the compilation of this document. Although the authors have attempted to provide accurate information in this document, the Trustcom Consortium assumes no responsibility for the accuracy of the information.

Information is subject to change without notice.

Mention of products or services from vendors is for information purposes only and constitutes neither an endorsement nor a recommendation.

Reproduction is authorised provided the source is acknowledged.

IBM, the IBM logo, ibm.com, Lotus and Lotus Notes are trademarks of International Business Machines Corporation in the United States, other countries or both.

Microsoft is a trademark of Microsoft Corporation in the United States, other countries or both.

SAP is a trademark of SAP AG in the United States, other countries or both.

'BT' and 'BTexact' are registered trademarks of British Telecommunications Plc. in the United Kingdom, other countries or both.

Other company, product and service names may be trademarks, or service marks of others. All third-party trademarks are hereby acknowledged.

Deliverable datasheet

Project acronym: TrustCoM

Project full title: *A trust and Contract Management framework enabling secure collaborative business processing in on-demand created, self-managed, scalable, and highly dynamic Virtual Organisations*

Action Line:

Activity:

Work Package:

Task:

Document title:

Version:

Document reference:

Official delivery date:

Actual publication date:

File name:

Type of document: Report

Nature:

Authors: Theo Dimitrakos (BT), Michael Wilson (CCLRC), Alvaro Arenas (CCLRC), Adomas Svirskas (CCLRC), Jochen Haller (SAP), Pablo Giambiagi (SICS), Jürgen Doser (ETH), Lutz Schubert (HLRS), Emil Lupo (IC),

Reviewers:

Approved by:

Version	Date Date	Sections Affected
V.1	30/06/05	All
V.4	31/08/05	introduction

Executive Summary

This document describes a framework to address issues of trust, contracts and security within dynamic virtual organisations supported by service oriented architectures including Web services and the Grid. A framework is a way to approach the problem, combining models to conceptualise the problem and possible solutions, an architecture that can support the solution, software methods & tools to implement the solution, and guidelines to address issues not addressed within implementations.

The emergence of Web services technologies support dynamic liaisons between organisations to undertake business plans, where no single organisation has either the required resources, or the desire to manage the risks involved. Such virtual organisations (VO) are not created as legal entities in their own right, but through co-operation agreements between organisations to aggregate resources and distribute risks. Within VO there are risks to each participant arising from the process in which they are involved as well as the operation of the other members of the VO. These risks can be mitigated through:

- clear definition of the business process in which they co-operate
- reputation management systems to evaluate potential partners
- contracts and service level agreements between partners to limit liability
- the derivation of security policies from those contracts that can be implemented and managed across the architecture supporting the VO
- accounting arrangements to ensure governance of the contracts.

However, there remain residual risks that are not mitigated by these measures, and for which trust is the main risk management measure. To increase or preserve trust levels throughout the lifecycle of the VO further actions can be identified. The framework proposed here addresses risk management for VO using this approach (excluding financial accounting which is addressed elsewhere).

It is proposed that the implementation of IT support for the framework be based upon open specifications of Web services technologies in order to maximise the potential range of partners in a VO, and to use market competition between suppliers to control implementation costs. However, existing and proposed specifications of Web services technologies do not currently interoperate with all others, nor are all specified features optimised, or even required for the proposed application. Therefore the framework proposes profiles of such web service technology specifications which are appropriate to the framework, its requirements and its application.

This document presents the first version of the Trustcom Framework, which is incomplete in many areas, but which does identify those areas for future work before the final version which is planned for release in January 2007.

Table of Content

1 Introduction to the Trustcom Framework 7
1.1 VO and the Trustcom Framework 9
2 Usage Scenario 11
2.1 Examples of Virtual Organisations in Collaborative Engineering 11
2.2 Examples of Virtual Organisations for Next Generation Service Providers 12
3 Technology Interoperability Analysis 15
3.1 Analysis of Enabling Technologies 15
3.2 Analysis of Technologies for Contract and Service Level Agreements 18
3.3 Analysis of Technologies for Security 19
3.4 Analysis of Technologies for Trust Management 21
3.5 Analysis of Technologies for Business Processing 22
4 Technology Profiles 25
4.1 Introduction to Profiles 25
4.2 VO Management (VOM) – CCLRC 28
4.2.1 VO concepts and innovative functionality 28
4.2.2 VO Management concepts and innovative, added-value functionality 31
4.3 BPM - SAP 33
4.3.1 BPM concepts 33
4.3.2 BPM Technology Standards 35
4.3.3 BPM planned work 37
4.4 SLA - SICS 37
4.4.1 Introduction 37
1.4.4.1 Background 38
1.4.4.2 Scope 38
4.4.2 SLA Identification 38
4.4.3 Parties 38
3.4.4.1 Signatory Parties 39
3.4.4.2 Supporting Parties 39
4.4.4 Obligations 41
4.5 T&S - UoK 42
4.5.1 Introduction 42
4.5.2 Reputation Management Infrastructure 44
4.5.3 Trust and Security Standards 45
3.4.5.1 PEP to STS-V 45
3.4.5.2 PEP to PDP 46
3.4.5.3 Reputation Service to Reputation Service 46
3.4.5.4 Trustor/Trustee to Reputation Service 47

3.4.5.5 VOMS to Reputation Service 47
3.4.5.6 Trust Negotiation 47

4.6 Policy - ICSTM 48
4.6.1 Introduction 48
4.6.2 Relationships: a structuring principle 49
4.6.3 The Policy Specification Language (2Ponder) 51
4.6.4 The enforcement model 54
4.4.6.1 Relationships 54
4.4.6.2 Obligation Policies (ECA model) 55
4.4.6.3 Access Control Policies (Authorisation & Delegation) 56
4.4.6.4 Inside the PDP 56
4.6.5 Policy Analysis Issues 56
4.6.6 Standards and Profiling 57
4.6.7 Conclusions 58

EN/VO Infrastructure - HLRS 58
4.6.8 WS Resources 59
4.6.9 Manageability 60
4.6.10 Notifications 60
4.6.11 Additional issues 60

5 Future Plans 62

1 Introduction to the Trustcom Framework

> The use of the internet is beginning to penetrate the daily life of citizens. A precondition for it to become more widespread is that it proves worthy of our trust. Security, privacy protection, property protection and general governance of the sector are indispensable for building citizens' confidence in the Information Society.
>
> *EC Communication, Nov 2004:Challenges for the European Information Society beyond 2005*

This report describes a framework to address issues of trust, security and contract management within dynamic virtual organisations supported by service oriented architectures including Web services and the Grid.

A framework is a way to approach the problem, combining models to conceptualise the problem and possible solutions, an architecture that can support the solution, software methods & tools to implement the solution, and guidelines to address issues not addressed within implementations. The Trustcom framework is a substantial development of a starting point proposed by Dimitrakos[1] for a service oriented trust management framework. A framework can be contrasted with a more precise methodology which defines the process of developing a solution, from the generation of requirements through refinements to implementation and testing. The framework is only a step on the way to such a precise development approach, but also addresses wider concerns outside of software – e.g. legal issues, business models etc..

Virtual organisations (VOs) are created when consortia of legal entities wish to work together to produce a product, provide a service or tender for a contract, but do not wish to either have one contracted party to which the others are subcontracted, or to create a new legal entity which they jointly own. VOs can be created quickly, and undertake their role for a very brief period of time, or exist for the longer term. Virtual organisations (VO) are not created as legal entities in their own right, but through co-operation agreements between organisations to aggregate resources and distribute risks. Within VO there are risks to each participant arising from the process in which they are involved as well as the operation of the other members of the VO. Business risk has been defined as "the threat that an event or action will adversely affect an organization's ability to achieve its business objectives and execute its strategies successfully" (The Economist Intelligence Unit 1995). Assessment and control of business risks has become increasingly important in recent years due to changes in information technology and related developments. Information technology has reduced the time available to react to environmental change, streamlined and altered the design of business processes, and changed the optimal form of organization. These developments have led to a de-layering and downsizing of businesses, resulting in fewer employees devoted to control activities, while also supporting a balance between the operational management of processes and the strategic management of organisational objectives. As these changes have reduced traditional controls over information and the safeguarding of assets, they have also laid the foundations for the rise

[1] Theo Dimitrakos (2003) **A Service-Oriented Trust Management Framework,** Proc. *International Workshop on Trust in Agent Societies, Melbourne, Australia, 14-15 Jul 2003,* Trust, Reputation and Security: Theories and Practice

of virtual organisations, and for the use of information technology to manage the risks that they give rise to.

Available analyses of VO reviewed in the Trustcom State of the Art [2] suggested that the highest impact risks identified with significant probability of occurrence could be mitigated through the following measures:

- clear definition of the business process in which they co-operate
- reputation management systems to evaluate potential partners
- contracts and service level agreements between partners to limit liability
- the derivation of security policies from those contracts that can be implemented and managed across the architecture supporting the VO
- accounting arrangements to ensure governance of the contracts.

The framework is constructed by addressing the first four of these measures through specifications for:

- a set of semantically well-founded **concepts** and relationships for describing and reasoning about trust and security in dynamic virtual organisations. This forms the meta-model of the TrustCoM framework;
- an abstract **architecture** reflecting these concepts, and providing a flexible structure and organising principles for systems based on the framework;
- **specifications** extending existing or defining new interoperability standards of services and protocols. (New interoperability standards will be defined when existing approaches cannot provide an extensible basis to support the TrustCoM framework).

The first of these specifications defines concepts and models that support trust, security and contract management. This work is described in a series of deliverables covering the core concepts[3], the business models[4], the economic issues[5] and the legal issues[6].

The second of these is defined in the Trustcom Architecture which has been defined a separate deliverable[7]. The architecture is divided into a set of components which are used both to structure the architecture and implementation of tools to support the framework, but also to structure the conceptual modelling and profiles extending existing specifications, that are built upon in the implementation. The overall components are illustrated in Figure 1 below.

---

[2] Nilufer Tutuk, Emil Lupu (2004) D2 – Trustcom State of the Art, http://www.eu-trustcom.com/DownDocumentation.php?tipo=docu&id=163

[3] D16 The Trustcom Conceptual Models, July 2005

[4] DA Business Models for Virtual Organisation Management and Interoperability, March 2005

[5] D14 WP8 Economics, Business and Social, July 2005

[6] D15 Report on Legal Issues, July 2005

[7] D09 Trustcom Reference Architecture, Lutz Schubert, August 2005

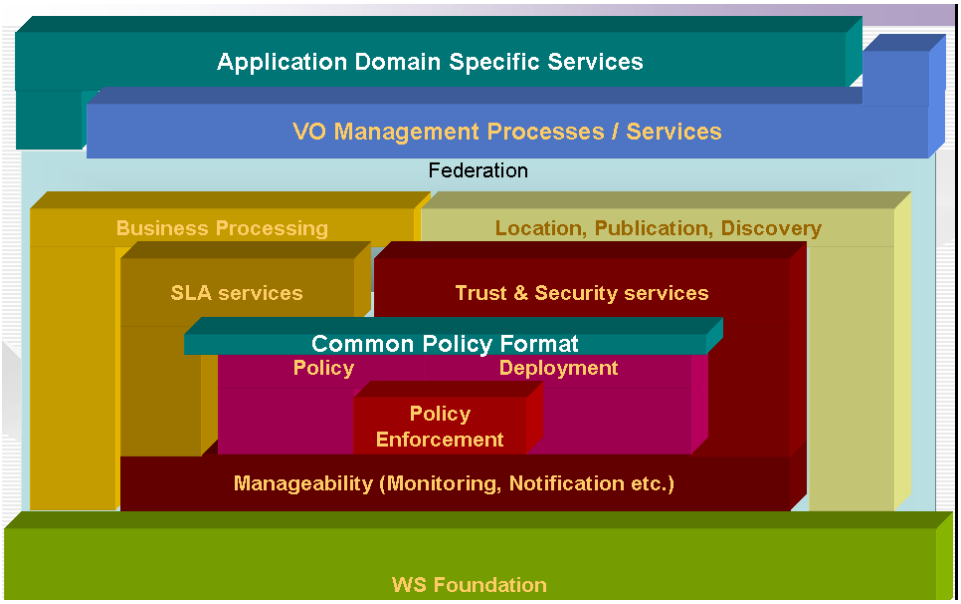

Figure 1: Trustcom sub-systems

However, there remain residual risks that are not mitigated by these measures, and for which trust is the main risk management measure. To increase or preserve trust levels throughout the lifecycle of the VO further actions can be identified. The framework proposed here addresses risk management for VO using this approach (excluding financial accounting which is addressed elsewhere).

1.1 VO and the Trustcom Framework

VOs can be established by a consortium or VO agreement which outlines the legal framework for the co-operation, within which specific Service Level Agreements can be produced to detail each service provided by a legal entity within the VO.

When a VO is formed it will have a view of the business it is conducting, and the roles in the business process that need to be filled by each potential VO member. Therefore the VO must discover potential members who can meet those role definitions, negotiate their agreement to the VO agreement, and specific SLA's for the role. Within Trustcom, it is assumed that the descriptions of potential members, and the services to be provided are all available as Web or Grid services, open to automated resource discovery, negotiation and SLA agreement. One of the factors involved in the selection of VO members will be their previous reputations both to undertake roles defined in the VO business process model, but also to operate under VO agreements and SLAs, and even, their litigiousness.

Once the VO is operating, each participant must open up its internal ICT infrastructure to the other VO members in as much as they require it to undertake their roles in the VO. Consequently there are security issues for each VO member concerning the authenticated identity of employees of their own and other VO members and authorisation issues of access to data and services throughout each organisation. While a VO is operating, the set of Web or GRID services brought together to achieve its business process model must be monitored and managed to ensure that the cost, time, security and quality measures stated

in each SLA are being met, and when they are not, that appropriate actions are taken. The VO software embodying the resolution of these issues must also resolve the standard tradeoffs of distributed computing between orchestration and choreography as methods of service composition. The VO software must conform to open standards to permit the software interoperability required for disparate organisations themselves to interoperate as required to bring about dynamic virtual organisations.

When a VO has completed its activity, it must then be terminated to minimise future risks from liability and exposure to security breaches, and ensure the appropriate accounting of expenditure and income and distribution of profit or loss between the participating organisations.

This vision encompasses the business process model (BPM) of a VO, the roles defined in it, the discovery of organisations to fulfil those roles on the basis of reputation, the legal agreements between selected organisations to fulfil their roles, the establishment of secure and reliable composition of Web or GRID services to enact the business process models, the monitoring and management of the performance of members, and the decomposition of the VO resulting in minimal outstanding risks. The relation between the executable BPM, SLA management and security concerns all result in policies which must be deployed and monitored during performance to reduce risk within an international legal framework. Therefore each of these components BPM, discovery and negotiation, reputation management, SLA management, security policies, legal framework needs to be analysed, modelled and incorporated into a software system to host dynamic virtual organisations where risks and the trust required to offset them can be quantified and judged by business decision makers.

2 Usage Scenario

Trustcom has undertaken analyses of many VO scenarios and many use cases for each scenario as a basis for the Framework. These are described in public deliverables on scenarios[8], but two are briefly summarised here to present examples of the class of VO being addressed, and the benefits of applying the Trustcom Framework.

2.1 Examples of Virtual Organisations in Collaborative Engineering

The development, production and support of modern products such as ships, aircraft etc are highly complex processes that often involve great risk. Principal risks include technical complexity (both in the complexity of products and processes) and changing customer and market requirements. The ability to manage these and other risks is a distinguishing feature of competitive organisations in the engineering sector. A strategy for managing this complexity is to form partnerships or Joint Ventures (JVs) in order to exploit new markets and opportunities through Collaborative Engineering (CE). In a JV partners focus on particular aspects of the product through its lifecycle, enabling more focus on core business capabilities. Emerging technologies such as web and grid computing may facilitate the evolution of JVs into Virtual Organisations (VOs), where organisations quickly come together to share resources without requiring the development of new facilities, and systems- a common feature of JVs at present. The CE scenarios described here attempt to cover most of the phases of the product lifecycle within a CE VO through development, production and in-service product upgrade.

The first of the scenarios focused on the collaborative design of a product in order to win a customer contract. The VO included principal partners who supply major subsystems collaborating together to deliver a design that meets customer requirements and will lead to the award of a production contract. Customer requirements are negotiated and constantly refined and re-negotiated based on the results of the collaborative design activities. The VO also includes other collaborators who provide technical capabilities from High Performance Computing through to highly specialised engineering analysis services. Capabilities that currently reside within monolithic enterprises, and which are expensive to support, are outsourced and yet can still be integrated closely with the product development process within the VO. The benefits to the VO from this sharing of design data are improved understanding of customer requirements and reduced risk by more extensive design investigations using external specialist services. The interaction between partners' security policies that control access to the design data and collaborative agreements, defined at the business level, is an important issue in this scenario.

The second scenario focused on the management of complex engineering processes associated with the production phase. These processes span enterprises, potentially in different countries or economic zones and are possibly subject to export restrictions. Each partner providing a major sub-system (eg, engine, fuselage of a civil aircraft) also has a

[8] Trustcom deliverable D3, Case Study Scenarios, 9th June 2004

network of component suppliers and logistics support. These production processes are complex and may require re-scheduling in the case of delays in deliveries, etc. This may necessitate the sharing of process information (so-called 'process visibility') across the VO, entailing flexible security systems that control access to internal information. Other issues include the identification of trustworthy suppliers who can deliver components to time and specification in accordance with service level agreements.

The third scenario looked at the upgrade of an in-service airliner to include an in-flight entertainment system. The VO reconfigures to admit a new sub-system provider responsible for delivering this system. The new partner is also a member of other VOs, possibly competitors to the current VO. An important business decision concerns the trustworthiness of the product data- the new partner will need to establish the completeness and accuracy of the product data before it can plan its activities. A possible consequence here is the possible withdrawal of the partner from the VO if this data is of poor quality and poses unacceptable risks in delivering to contract. The new partner also wishes to monitor the in-service performance of the system using the customer's operations data. The security system of the new partner must be trusted by the VO to be sufficient to not allow sensitive product data to be accessible to its competitors; it must also be trusted by the customer to not disclose operations data to its own competitors.

In summary, the three scenarios have highlighted the importance of effective and flexible security system for building confidence in the extensive and more integrated collaborations that VOs offer over conventional JVs. The security policies should also be correlated both with the collaborative agreements established between partners at the business level and with agreements established within other collaborations as well. The benefits from an effective security and contract management framework are the ability for engineering collaborations to be quickly reconfigured in order to expose the assets that need to be shared to achieve the business goal. Service level agreement monitoring is important for ensuring that suppliers (of components, services etc) perform according to contracts.

Benefits here possibly include the automation of processes between clients and suppliers that are usually repetitive. Finally, trust frameworks are required for supporting collaborations. The first of these concerns managing the reliability and trace ability of engineering data, ensuring that greater confidence can be given to it and that it can be relied on in major engineering tasks. The second of these Trust frameworks should facilitate the search for new partners/suppliers of components or services that were previously unknown to the VO. This should include some assessment of the trustworthiness of the security systems and its security policies.

2.2 Examples of Virtual Organisations for Next Generation Service Providers

We are interested here in VOs that are formed through ad hoc aggregation of component services offered by different 'real' service providers. Increasingly, enterprises are using web services and related technologies to provide their customers, suppliers and partners with direct access to their services and business processes. Motivations include reducing costs and speeding up processes through automation. However, the vision behind the web services / service oriented architecture revolution is that distributed applications can be assembled as needed by connecting together pre-existing services. Selection of the

services to use takes place through a 'discovery' process. As well as connecting the services together into a supply chain capable of fulfilling a customer order, the business process of the enterprises involved must also be interfaced. Furthermore, contracts need to be agreed establishing the mutual rights obligations of the participating service providers. When connections at the these three levels can be established on demand, we can truly say we have an and hoc dynamic VO.

We are already seeing services being 'disaggregated', that is, in addition to offering 'complete' services, simpler constituent services are offered separately. Other organisations can then make use of these constituents in combination with their own service elements to offer composite services to their customers. Motivations for disaggregation include regulatory / anti-trust factors, advantages arising from focus on core competences, business agility (ability to launch new services / enter new markets rapidly), a desire on the part of the individual SPs to retain the advantages of small scale (or conversely to avoid the overheads and inertia of large organisations. New services may also be created specifically for use as constituents of larger services offered by other enterprises. This could offer opportunities for specialist start-up companies to enter a market. Benefits of dynamic aggregation include provision of services that are precisely tailored to a specific customer need. The need to offer a wide range of tailored services could arise from a wide range of preferences or requirements among the targeted customer base, or because the specifics of the service depend on the circumstance of the customer, e.g. current location, the task currently being undertaken, and other context specific variables. The ability to participate in dynamic VOs greatly increases the range of services a provider can offer to its customers, and also the number of end-customers it can reach indirectly via partners.

Five such 'Aggregated Services' (AS) scenarios have been defined and analysed as part of the TrustCoM problem definition: The first takes uses one of the ISTAG Ambient Intelligence scenarios: 'Maria the Road Warrior'. This follows a near-future business traveller through her day, during the course of which she makes use of various services provided collectively by the many devices and systems making up the pervasive computing environment. We can see these services as being provided by virtual organisations forming in response to Maria's needs. The second scenario was similar, but less futuristic. A business traveller in a foreign city wants something to do in the evening. He calls into a WLAN-equipped café, and requests a personalised multi-media city tour guide to be shown using a combination of his personal computing equipment and the café's facilities. This service is provided by a VO consisting of the café, content providers / aggregators, the users 'home' service provider, and other network operators. The detail of the scenario focuses on the use of trust service providers representing the interests of the various parties.

In the third scenario, a small software company wants to bid for a new contract, which under normal circumstances would be beyond the scope of its resources. The company has involved itself in a network of similarly placed small companies and is able rapidly to form a VO to compete with larger corporations in the development of new technologies. It is interesting to note that analogies can be drawn between this scenario and the TrustCoM project itself, raising the possibility that TrustCoM could be used as a test-bed for its own technology developments. The fourth scenario concerns experiential e-learning that may take advantage of a combination of semantic-based content selection, personalization of learning activities and integration with high-performance capabilities for virtual

experimentation. Here, VOs providing highly personalised educational services are formed within a community of service, content and resource providers. A learner is assisted in defining a personalised training session, which is then enacted by the VO for the benefit of the Learner. The fifth scenario examines how a national or international incident (e.g. an environmental crisis) could be handled through collaboration among organisations that are selected according to context and must agree on a way of sharing and managing resources.

As the resources and services are owned and managed separately by these organisations a decentralised command and control system will be advocated to synchronise their efforts and their use of the resources in the most efficient way.

In summary, the five scenarios have highlighted that dynamic VOs inevitably incur a management overhead compared to real organisations, and indeed to static VOs (formal consortia). There is a requirement for additional services to provide the glue that enables the VO to function as a viable entity e.g. to provide overall coordination of activities while retaining flexibility. We expect that these services can be defined in such a way that they are basically independent of the particular application domain. Furthermore, there is a requirement for services to replace the trust inherent in operation within an integrated real organisation (trust in colleagues even when not known personally, trust in procedures and processes, etc.), and the trust between customer and an established service provider with a clear legal identity and brand / reputation. This last class of service is a main ingredient of the TrustCoM Framework. Without such a framework, it is likely that enterprises will judge that the risks in participating in dynamic VOs will out-weigh the benefits. Similarly, end customers will be reluctant to buy from dynamic VOs. It should also be recognised that there are substantial commercial opportunities for enterprises offering the trust, security and contract management services instantiating the TrustCoM framework. The TrustCoM project will prototype implementations of potentially useful classes of service, drawing on the scenarios mentioned above for requirements.

3 Technology Interoperability Analysis

TrustCoM undertook a technology interoperability analysis as the first stage toward the definition of TrustCoM profiles. Such analysis consisted of the following steps:

- Defining an abstract scenario, which is as abstraction of aggregated services and collaborative engineering scenarios, that covers the most relevant cases of both scenarios.
- Performing a requirement analysis exercise, highlighting trust, security and contract management requirements.
- analyse the base-line open standard technologies useful in implementing the abstract scenario, technologies selected from the State of the Art deliverable D2.

This section summarises the main results of the technology interoperability analysis. The complete analysis is available in internal deliverable ID1.2.5.

Technologies have been grouped according to the thematic areas of the project: policies and security, contract management and business processes. We start the technology description with some Enabling Technologies that could be useful in the project such as Web services and Semantic Web. The results of this study have fed other workpackages. For instance, parts of these results were presented as an addendum to WP10 deliverable on the State of Art, and have influenced work in WP3, the TrustCom architecture.

3.1 Analysis of Enabling Technologies

The spectrum of enabling technologies available for TrustCoM broadly divides into the following categories: Web Service specifications and their infrastructure support, grid technologies, semantic web and ontology based techniques, and tools and platforms for implementation.

TrustCoM has made an early commitment to Web Service standards (WS-*) and associated technologies. This constitutes at the same time an opportunity and a major source of risk for the project. It is an opportunity because in the current state of development, web services seem to be the technology of choice for interoperability and collaboration across heterogeneous implementation domains. It is a major source of risk, because beyond the basic SOAP/XML-RPC communication mechanisms many of the more advanced specifications are still in the draft phase. The specifications are sometimes overlapping and occasionally incompatible as widespread agreement over the set of functionalities that each standard should cover has not been achieved. The implementation status of a number of the specifications is unknown. IBM and Microsoft are the most active players in this arena but their software packages that implement these emerging specifications are frequently updated, incomplete in some respects and the vendors classify the implementations as technology previews.

Of the various standards, SOAP, WSDL, XML Schema, WS-Addressing and WSReliableMessaging constitute the basis for communication across domain boundaries. Together they provide the means to describe the functionality of Web Services, address them in a transport layer agnostic fashion and exchange messages reliably between the various components. In addition WS-Addressing is relied upon by many other specifications. Services need to be discovered and bound to dynamically. Whilst UDDI provides adequate means for registering and discovering services and should be used in the project, it is likely that standard implementations will need to be extended to accommodate the additional information about a service used within TrustCoM such as SLAs associated with a particular service, trust level and reputation.

Event-based asynchronous notifications play a particularly important role within a VO as changes of state need to be communicated to a potentially large and dynamically changing set of components. Two specifications WS-Eventing and WSNotification would form good candidates for this purpose. Both are robust specifications that define core features for event-based systems, although they lack some state-of-the-art properties. Additionally, the specifications overlap in scope and are currently incompatible because they target slightly different applications. WS-Eventing is a basic and general-purpose publish-subscribe event-based system. WS-Notification is aimed at management of resources and considered to be used in conjunction with WS-RF. WS-Notification also provides more sophisticated means of managing subscriptions, brokering and topic based dissemination of events. Although, the two specifications are expected to merge at some point in the future, this is not likely to happen during the first phase of development in the project.

To support management of web services and resources using web service-based protocols both WS-Resource (and on tom WS DM MUWS/MOWS) as well as the WS-Management specification (including WS-Transfer and WS-Enumeration) are an emerging effort for getting and setting properties of services. Depending on the requirements of VO management and the maturity of available tools, TrustCoM may use these specifications to perform management operations. WS-Resource, WS-Management (including WS-Transfer and WS-Enumeration), WSDM or a minimal self-defined service interface will be used for management of Web Services and associated resources as well as implementing adaptation.

Semantic Web technologies and in particular OWL-S can be used for describing a number of ontological structures related to a VO. In addition to representing structures such as role hierarchies and trust domain OWL-S also permits to publish semantic information related to a web service as well as process models explaining the dependencies between message exchanges. Such models are particularly useful in the cases where mutually intelligible vocabularies of terms for data and process descriptions need to be established between the participants in a VO. However, the expressiveness of OWL-S for web service sharing and integration in a VO context has not been proven yet. In consequence, TrustCoM should focus in a first stage on defining and implementing a working infrastructure for the establishment, evolution and enactment of VOs assuming a single notation for the specifications. Provided this first step is addressed successfully, a generalisation to

the use of general ontologies for describing terms and establishing mutually agreed vocabularies can be undertaken in a second phase.

Grid based systems exhibit a degree of similarity with Virtual Organisations and sometimes adopt the same terminology. In particular aspects such as service composition service discovery and simple aspects of SLAs have been implemented and demonstrated. However, Grid-based systems remain open environments focussing on the sharing of resources and distribution of computational tasks. Therefore, they do not address some of the more complex problems related to the recursive composition of VO structures or the use of business processes. Although, a number of security issues have been addressed they also do not cater for recursively-composed structures and do not account for different trust relationships between the participants of a VO structure. From an implementation point of view Grid-environments have been largely monolithic up to now. There are numerous dependencies between the various elements of the frameworks making it difficult to reuse particular tools or components in isolation from the rest. There is substantial expertise within the TrustCoM consortium on building Grid environments and this expertise will be used in the first phase to identify particular tools and techniques that could be re-used within the project.

As a conclusion of the analysis of enabling technologies:

- SOAP, WSDL, XSD, WS-Addressing and WS-ReliableMessaging should be used as baseline. Extended UDDI should be used for service discovery. WS-Notification and possibly WS-Eventing should be used for Monitoring and Event dissemination.
- WS-Resource, WS-Management (including WS-Transfer and WS-Enumeration) or a minimal self-defined service interface should be used for management of web services and associated resources as well as implementing adaptation
- A number of Grid-services and algorithms could be used, however they need to be extracted from the entire grid-framework and used in isolation.
- Although a number of ontology techniques and in particular OWL-S could be used, these would add an additional level of complexity to the project.

As a result the analysis, the following is recommended:

- The project needs to investigate and choose a set of tools and platforms as basis for development. If a particular standard is not supported by existing implementations compliance with that particular standard needs to be abandoned. Attempting to provide interoperable implementations for standards where they do not exist constitutes a major development effort beyond the effort available in the project.
- No more than two platforms need to be selected for implementation. Each platform should have a well-defined set of implementations of the WS-* standards on which it will base any further development.
- Identify if any Grid-based implementation elements can be isolated and reused.

- Use of semantic web technologies and in particular ontologies should be delayed until the basic infrastructure is implemented deployed and tested on a simplified version of the scenarios.

3.2 Analysis of Technologies for Contract and Service Level Agreements

Work on Service Level Agreements (SLAs) has been developed by the network and systems management community in order to cater for the specification of the Quality of Service (QoS) parameters characterising the provision of network connectivity services. This work has evolved into general frameworks for the characterisation of application level services and more recently business services. Most of the solutions proposed in this area provide the means for: specifying SLAs and associating them with the WSDL services concerned, discovering and locating services based on profiles of QoS that can be delivered for those services, defining simple negotiation protocols for negotiating QoS parameters, and monitoring the compliance with the SLA objectives (including monitoring and metric definition).

However, the extent to which these features are supported varies greatly amongst the different SLA solutions proposed. Probably the most concrete framework that is likely to provide a solid foundation for TrustCoM is WSLA, which in addition to specification and structuring of SLA agreements also provides detailed monitoring aspects including an extensible framework for metric definition. The other framework of particular interest is WS-Agreement. Originating initially from the OGSI framework, and a good example of how Grid platforms evolve towards a more open web service environment, WS-Agreement caters for the discovery of services including SLA retrieval and negotiation and is compliant with the other WSRF specifications. WS-Agreement is however a relatively new specification.

ebXML Trading Party Agreement and Collaboration protocol Agreement would also be an alternative. However, their specifications and implementations are tightly coupled to the other ebXML specifications, which do not seem to integrate well with the other web service specifications.

The conclusions of the analysis in this area are the following:

- The contract and service level agreement framework in TrustCoM can draw heavily on both the WSLA and WS-Agreement specifications and their implementation support. The aim is to design a framework that caters for both monitoring and enforcement aspects as well as service location and negotiation.
- There are not many studies examining in detail how trust and reputation information is to be used in conjunction with contracts, SLAs and their associated processes.
- Although, some of the frameworks mentioned above present some features for reacting to SLA violations, this is one of the areas in which TrustCoM could bring a significant contribution.

The recommendations for the development of the project in relation to this thematic area are:

- An explicit conceptual model for supporting agreements at both business and service level needs to be developed based on a conjunction of WSLA, WSAgreement and relevant concepts from the BCA architecture.
- The development of this conceptual model needs to devolve significant efforts to two aspects: on one part, the impact and use of trust and reputation relationships in service discovery, SLA negotiation and enforcement phases; and another part, the handling of SLA violations in a more flexible form similar to business process descriptions.
- Finally, there is a need to identify which specific implementations or parts of implementations could be re-used within TrustCoM.

3.3 Analysis of Technologies for Security

As mentioned before, Security aspects of a VO framework span a large number of concerns that broadly divide in the following categories: Access Control, Authentication, Secure Connections, Information Disclosure and Adaptive Security. These will each be addressed in turn in the paragraphs below. Overall security and policy are not only a substantial part of TrustCoM but one where the consortium has considerable expertise.

Access Control Models are well understood within a single administrative domain and new concepts such as Role Based Access Control are increasingly appearing in main stream products. Authorisation policies are used in a number of different frameworks (Ponder, Permis, SPKI, etc) and standards (XACML). Despite apparent differences between the specification languages their functionality is broadly similar. Their enforcement is sometimes different, in particular when applied in distributed environments but the advantages and disadvantages of the various solutions are again well understood. However, distributed access control within environments that cross-domain boundaries remains fundamentally an open research problem. Grid environments have attempted to address these issues in a number of platforms (Akenti, VOMS, CAS, etc.) however the assumptions on which these models are based are too restrictive for VO enforcement. In particular, most grid-platforms are concerned with access control to resources by distributed tasks and do not allow for recursively composable VOs in federated structures (i.e., a Grid is not itself a VO that can participate in higher-level VOs). One common characteristic across all platforms is however the increased usage of arbitrary security tokens to convey relevant security information. As domain boundaries are crossed, local identity loses any meaning and access control decisions are made based on properties that the requestor proves he possesses. These properties may include its role, qualifications and other attributes as well as privileges he/she holds or that have been delegated to him/her. This evolution is also evidenced in the more recent web-service standards such as WS-Trust, SAML and WS-Federation. The latter, in particular, focuses on the exchange and use of such tokens across domain boundaries. WS-Trust and SAML overlap in scope.

Authentication, and in particular authentication based on identity, becomes then a particular case of the more general token based framework described above. Recent studies and standards have particularly focussed on Single Sign-On systems such as Liberty Alliance and Shibboleth. Both of these overlap in scope with WS-Security, WS-Trust, WS-Federation based standards but tend to be less flexible (e.g., lack of support for “active” requestors), focus on identity management alone and rely on SAML for communication of information and SSL as the underlying secure transport protocol.

The WS-* series of specifications provides solutions for message integrity and confidentiality that are specifically tailored the SOAP messages exchanged with web-servicies. These are WS-Security and WS-SecureConversation and have been designed to work in conjunction with the other specifications from the series.

Invariably, access control, authentication and secure connections rely upon the security services being appropriately configured for communication and interoperation. This is achieved through policies, and thus WS-Policy, WSSecurityPolicy, WS-PolicyAssertions and WS-PolicyAttachement have been introduced. These specifications aim to provide an interoperable format for policies which can then be embedded and exchanged as secure tokens. Although originally conceived for the local configuration of services the common aspects have been generalised and some advocate the use of these standards for any kind of policies.

The ability to control information dissemination and disclosure including providing restricted access to sub-parts of a document is a long standing research problem. Although expertise exists within the consortium to address these issues, this would entail using a disproportionate amount of effort. In would therefore be appropriate to delay these issues until later on in the project when a security infrastructure is in place.

As often in the past most of the existing work focuses on security mechanisms rather than on the processes required for managing them or adapting their behaviour. Although there are several specifications on how to define policies there is little work on which policies to apply in which circumstances or how policies are dynamically changed in response to changes in context or trust. This issue is of particular relevance in a VO environment characterised by varying trust relationships and where the members and structure of the VO may change. Consequently this is arguably the most important challenge that the TrustCoM security framework will need to address. Even the ability to tackle this challenge is only possible because of the integration with Business Processes, Contracts and Trust.

As result of the analysis in this area, we can conclude:

- WS-Policy, WS-SecurityPolicy, WS-Trust, WS-SecureConversation and WSFederation form a coherent group of standards that is explicitly focussed on open web-service environments.
- However, several issues need to be addressed in order to combine them in a security infrastructure that satisfies the needs of VO Environments.

- Although aimed at being general, WS-Policy and WS-SecurityPolicy remain attached to low level service configuration. Therefore XACML or similar policies may prove more useful ad higher levels of abstraction for use within a VO.
- Information disclosure and dissemination remains a significant issue but it is unlikely that the project will be able to address this, at least in the first phases of development.
- Realising adaptive security represents the most important research challenge. However, little work exists in this area that can be directly leveraged.

As a recommendation in the project,

- The efforts on security and policy should proceed along two parallel but sidestepped tracks: the definition and implementation of a suitable model for authentication, access control and secure connections within a VO environment; and the design and development of the adaptive security model. The starting point for investigation should in particular concern adaptation as a function of trust.
- There is a need to identify early in the projects the implementation available for the WS-* security standards, their degree of maturity and interoperability. The lack of adequate implementation toolkits is a significant risk which would require significant re-development of the model to compensate.

3.4 Analysis of Technologies for Trust Management

Technologies for trust management can be broadly divided into two categories: those that focus on trust aspects of a security infrastructure in particular with regards to the authentication of users or disclosure of information; and general frameworks for trust management that focus on trust analysis, quantification and trust services.

The technologies for trust from a security perspective are relatively well understood in particular when relating to PKI infrastructures. In addition, there are also a number of emerging studies on trust negotiation i.e., the incremental disclosure of security relevant information such as credentials and requirement for access although further studies are needed in this area. The latter have also been subject of a number of studies but there is little consensus on how to define, manage and compute trust based on an infrastructure of trust services.

From the analysis in this area, we can conclude:

- Trust management models need to be developed in TrustCoM together with a supporting infrastructure. Considering the lack of consensus on trust management at the research level it is unlikely that in a VO setting the participants will adopt similar trust metrics or computational models.
- There is some initial work on trust negotiation in terms of incremental disclosure of credentials and requirements. This work can be extended and included in the TrustCoM framework.

- PKI and other security aspects of trust are sufficiently well understood to be used as examples in the over all trust management framework and evaluate a subset of its expressiveness.
- Trust plays a very important role when establishing business partnerships and collaboration and maintaining them over time. However more evidence has to be acquired to demonstrate how trust is incorporated both in the contract content and, foremost, in the operational business processes.

It is then recommended in the project:

- The trust management infrastructure needs to be agnostic to specific trust metrics or computational models. In addition protocols for negotiation and exchange of trust information need to be developed.
- Security aspects of trust are well understood and should be treated like a specific case in the overall trust management framework rather than as a distinct issue. This will also enable us to test (to a certain extent) the expressiveness and use of the framework.
- Given the current state of the art, it seems reasonable to begin with the development of a trust management model and infrastructure. This should include: what trust services are provided at the VO level and how the information is aggregated from the participants, what specifications of trust metrics and computational models are needed to provide coherent trust information to the decision functions within the VO, how this information will be used as part of SLA management and negotiation, business processes and autonomic security enforcement.

3.5 Analysis of Technologies for Business Processing

Collaborative Business Processes are probably the best understood and defined of the analysed technologies. Indeed, the issues regarding executable collaborative business processes in the last few years have been more focussed towards standardisation aspects rather than basic research, as many software vendors and business integration consultants are using a wide spectrum of proprietary protocols. Standardisation allows addressing the problems of executable business process aggregation and collaboration across administrative domains that use proprietary solutions as well as outsource workflow control and implementation to third parties.

A number of specifications have been investigated including: WS-Coordination that defines the means to coordinate distributed actions during process runtime including agreement on outcome through the propagation of activity contexts WSTransactions that extends context information to include transactional capabilities for both atomic transactions (WS-AtomicTransactions) and long running business transactions (WS-BusinessActivity), BPML/WS-CI that focuses on the choreography of message exchanges starting at design time across multiple parties and BPEL4WS that provides the means to describe abstract and executable business processes in terms of their structure, control as well as offered and

invoked service interfaces. BPEL4WS and BPML/WS-CI have overlapping functionality, in particular for the business process specification although from different points of view. Whilst BPEL4WS relies on supporting Web Service standards such as the WS-Coordination model, which relies on the use of a single coordinator entity or a hierarchy of coordinators to control the execution of the workflow, WS-CI advocates a more loosely coupled choreography model with distributed control. Although our generic scenario includes a coordinator (the Orchestrator), in many cases there is not explicit requirement of a coordinator and WS-CI may provide some flexibility. Regrettably, development of the BPML/WS-CI has been abandoned with most of the concepts being integrated in a new specification, WS-CDL. The latter however, is still evolving and is not sufficiently stable to base the TrustCoM development upon it, at least during the first stage of the project. WS-CDL is also not catering for a collaborative business process choreography description capturing complex message exchanges across administrative domains, for instance n tendering and quotation processes.

Another option that has been investigated is the use of the ebXML-*series of specifications. However, these do not seem to integrate well with the other WS-* specifications since they advocate their own way of implementing messaging, service repository access, security, etc. It was therefore felt that these should not be investigated further.

In relation to business process area, we can conclude

- The most promising and stable approach to be used within TrustCoM is based on BPEL4WS/WS-Coordination/WS-Atomic Transaction. WS-CDL should however be monitored for further developments.
- Few, if any of the existing studies address business processes in conjunction with SLAs and none in conjunction with trust and reputation information for service selection and composition.

It is therefore recommended:

- Business Process Modelling and implementation should proceed based on the above-mentioned specifications and any available packages providing adequate implementations.
- As the business process infrastructure needs to be deployed for application level services the consortium should investigate further how the same infrastructure can be used to support the administrative and possibly adaptation processes inside the VOs.
- After an initial phase of defining and implementing the core business process functionality, the efforts should focus on two aspects: integration with the SLA infrastructure and leveraging the availability of trust and reputation for providing enhanced flexibility in the enactment of the processes especially across administrative domains.

4 Technology Profiles

4.1 Introduction to Profiles

The purpose of a Profile is to identify those features of one or more standard technology specifications that are required to allow effective or interoperable use of software in a limited range of applications.

Implementation of a profile by systems developers will improve interoperability among diverse systems and improve results within specific functional areas. What are sometimes considered as two related classes of profiles are considered here: firstly, profiles as a subset of a single standard to meet the functionality required by a specific application; secondly, profiles of multiple standards to achieve interoperability. However, since the second class can only be justified for a limited set of application functionalities, the justifications and structure of the first class also apply to them, therefore the two classes will be treated as one within this document, merely differentiated by whether they address a single or multiple standard technology specifications.

The usefulness of a profile is twofold. First, it is intended to define a core set of functionality and appropriate specifications to enable interoperable performance, especially when one system does not have detailed information about one or more other systems. Second, profile specifications provide the foundation for interoperability between systems outside of the primary jurisdiction of regional, national, state, local, and project groups when their individual profiles incorporate the main Profile specifications. The core functionality and specifications defined in a profile are intended to serve as a true subset of regional, national, state, local, or project profiles.

Each function is not an end unto itself but rather one step within a larger business activity. For this reason, a profile will usually be developed incrementally to incorporate richer functionality that can address additional applications. However, a profile will be defined for one or more Functional Areas which may still only require subset of the functionality provided by standard technology specifications.

The primary goal of a profile is to increase the semantic interoperability between disparate systems so that end-users can use them to perform defined functions and be confident that they have produced valid results.

A key component of any profile is the characterization of the types of functions required by users for the application area being addressed. To this end, a profile defines specific functions and how the semantics of those functions are to be expressed in the vocabulary of one or more standard technical specifications. Profiles do not prescribe local practices, however, semantic interoperability may be threatened by local decisions that serve local needs. Because a profile defines a core set of functions desired by users, implementers may use these functions to guide local decisions.

An prototypical example of a profile is the Bath Profile for the Z39.50 protocol specification[9] to be used for library applications and resource discovery [1]. This profile was developed from a set of local profiles generated by different universities and national groups. Each profile defined a set of functions required by their local library services and showed how the Z39.50 protocol should be used to perform those functions. The Bath profile is more generic than any of the previous profiles in that it aggregated the functions from them all together to address a wider range of library functions. However, even the Bath profile still only addressed how the Z39.50 protocol could be used to perform the functions required for library applications. There are other profiles of Z39.50 developed for cultural heritage data held by museums and galleries, and even for thesaurus navigation which address functions not considered in the library application.

In summary, a single technical standard specification can be developed to provide a larger set of functionality than is required for a set of functions in one application area. Indeed there may be ambiguous possibilities of different ways of implementing a function with a standard in for an application area. A profile states an unambiguous single way in which the set of functions required by an application should be implemented.

When considering not a single specification standard, but multiple standards a second class of profile becomes significant. Profiles provide implementation guidelines for how related Web services specifications should be used together for best interoperability. For example, WS-I[10] has finalized the Basic Profile, Attachments Profile and Simple SOAP Binding Profile. Work on a Basic Security Profile is currently underway within WS-I. In the Basic Profile, WS-I include a set of guiding principles for profiles which are also being followed in the Trustcom project::

No guarantee of interoperability
: It is impossible to completely guarantee the interoperability of a particular service. However, the Profile does address the most common problems that implementation experience has revealed to date.

Application semantics
: Although communication of application semantics can be facilitated by the technologies that comprise the Profile, assuring the common understanding of those semantics is not addressed by it.

Testability
: When possible, the Profile makes statements that are testable. However, such testability is not required. Preferably, testing is achieved in a non-intrusive manner (e.g., examining artifacts "on the wire").

Strength of requirements
: The Profile makes strong requirements (e.g., MUST, MUST NOT) wherever feasible; if there are legitimate cases where such a requirement cannot be met,

[9] UKOLN (1999) **The Bath Profile: An International Z39.50 Specification for Library Applications and Resource Discovery**, Release 1.0 - http://www.ukoln.ac.uk/interop-focus/activities/z3950/int_profile/bath/draft/stable1.html

[10] WS-I (2004) Basic Profile Version 1.1 - http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

conditional requirements (e.g., SHOULD, SHOULD NOT) are used. Optional and conditional requirements introduce ambiguity and mismatches between implementations.

Restriction vs. relaxation

When amplifying the requirements of referenced specifications, the Profile may restrict them, but does not relax them (e.g., change a MUST to a MAY).

Multiple mechanisms

If a referenced specification allows multiple mechanisms to be used interchangeably, the Profile selects those that are well-understood, widely implemented and useful. Extraneous or underspecified mechanisms and extensions introduce complexity and therefore reduce interoperability.

Future compatibility

When possible, the Profile aligns its requirements with in-progress revisions to the specifications it references. This aids implementers by enabling a graceful transition.

Compatibility with deployed services

Backwards compatibility with deployed Web services is not a goal for the Profile, but due consideration is given to it; the Profile does not introduce a change to the requirements of a referenced specification unless doing so addresses specific interoperability issues.

Focus on interoperability

Although there are potentially a number of inconsistencies and design flaws in the referenced specifications, the Profile only addresses those that affect interoperability.

Conformance targets

Where possible, the Profile places requirements on artifacts (e.g., WSDL descriptions, SOAP messages) rather than the producing or consuming software's behaviours or roles. Artifacts are concrete, making them easier to verify and therefore making conformance easier to understand and less error-prone.

Lower-layer interoperability

The Profile speaks to interoperability at the application layer; it assumes that interoperability of lower-layer protocols (e.g., TCP, IP, Ethernet) is adequate and well-understood. Similarly, statements about application-layer substrate protocols (e.g., SSL/TLS, HTTP) are only made when there is an issue affecting Web services specifically.

Requirements state the criteria for conformance to the Profile. They typically refer to an existing specification and embody refinements, amplifications, interpretations and clarifications to it in order to improve interoperability. All requirements in the Profile are considered normative, and those in the specifications it references that are in-scope (see "Conformance Scope") should likewise be considered normative. When requirements in a Profile and its referenced specifications contradict each other, the Profile's requirements take precedence for purposes of Profile conformance.

The scope of the Profile delineates the technologies that it addresses; in other words, the Profile only attempts to improve interoperability within its own scope. Generally, the Profile's scope is bounded by the specifications referenced by it.

The Profile's scope is further refined by extensibility points. Referenced specifications often provide extension mechanisms and unspecified or open-ended configuration parameters; when identified in the Profile as an extensibility point, such a mechanism or parameter is outside the scope of the Profile, and its use or non-use is not relevant to conformance.

Note that the Profile may still place requirements on the use of an extensibility point. Also, specific uses of extensibility points may be further restricted by other profiles, to improve interoperability when used in conjunction with the Profile.

Because the use of extensibility points may impair interoperability, their use should be negotiated or documented in some fashion by the parties to a Web service; for example, this could take the form of an out-of-band agreement.

The following sections describe the profiles that are being developed within the Trustcom framework to guide implementers.

4.2 VO Management (VOM) – CCLRC

The current stage of development of VO management has not specified profiles of existing specifications and how they should be used to support the Trustcom Framework. Therefore this section defines the advances over previous approaches tat have been designed, and concludes with a brief description of the specifications which will be included in future profiles as the project develops.

4.2.1 VO concepts and innovative functionality

This section outlines the aspects of VO management that TrustCoM innovations will help bringing about, emphasising some of the fundamental common characteristics that distinguish such emerging forms of VO from more traditional enterprise networks.

Virtual Organisations versus Enterprise Networks: A Virtual Organisation is understood as a temporary or permanent coalition of geographically dispersed individuals, groups, organisational units or entire organisations that pool resources, capabilities and information to achieve common objectives. Virtual Organisations can provide services and thus participate as a single entity in the formation of further Virtual Organisations. This enables the creation of recursive structures with multiple layers of “virtual” value-added service providers. The parties that form a virtual organization are typically part of a larger enterprise network of which a selection of partners is made. This phenomenon is known as “network activation” in VO modelling theory (See for example [11]). The entities in the Universe of such networks share some broad characteristics, e.g. belonging to the same economy or market sector, and their participation in the network indicates disposition to work together in a future market opportunity.

[11] Saabeel, W., Verduijn, T.M., Hagdorn, L., Kumar, K. (2002), A Model of Virtual Organisation: A Structure and Process Perspective, Electronic Journal of Organizational, Virtualness, 4: 1. 2002

A common VO life-cycle model: The following life-cycle model[1] emerged as a reference, although it was noted that not all of these phases would necessarily fit the specifics of all VO variants considered in the project.

- *VO Identification*. This phase involves opportunity identification, opportunity evaluation and selection.

- *VO Formation*. This phase involves partner identification, partner evaluation and selection, and partnership formation, including the binding of the selected candidate partners into the actual VO.

- *VO Operation & Evolution*. This phase is characterised by the controlled integration of the services and resources, offered by the VO partners in VO-wide collaborative processes leading to the achievement of shared business objectives. Membership and structure of VOs may evolve over time in response to changes of objectives or to adapt to new opportunities in the business environment. Changes of VO context may necessitate contract amendment or adaptation of policy and business process enactment.

- *VO Dissolution*. This phase is initiated when the market opportunity is fulfilled or has ceased to exist. The major decision processes in the termination phase include operation termination and asset dispersal.

Targeted Virtual Organisations: This emerged as a characteristic of VOs that are formed and exist for a *purpose* and in response to a *market opportunity* or in order to fulfil a *market demand*. Consequently, the VO focuses on a particular market segment or target group. Targeted VOs are characterised by *goal-specificity*, i.e. activities and interactions of their constituents are coordinated in order to achieve explicitly defined goals that provide unambiguous criteria for selecting among alternatives, and by *deliberate cooperation*, i.e. the structure of relations between VO partners and the services and resources they contribute is made explicit and can be “deliberately reconstructed”.

Dynamic Virtual Organisations: Membership and structure of such a VO may evolve over time to accommodate changes in requirements or to adapt to new opportunities in the business environment. While the VO as a collective collaborates towards a common objective, parts of the VO capacity and capabilities are owned by different independent partners, which have their own (partly overlapping, partly conflicting) interests. When the goal of a partner is met or the partner feels its own objectives no longer align with the goal of the VO, it can step out of the VO. Partners collaborating in order to perform a task in a phase of a VO may leave and join a different, potentially competing VO within the life-time of the former. The ability to self-organise is a key attribute of cost-effectiveness for dynamic VOs. A specific kind of dynamic VO are those that have the capability to unite quickly in order to exploit an apparent opportunity or common goal. We refer to this as *“on-demand formation of a VO”*. Dynamic VOs are particularly useful when faced with market incentives for responsiveness, dynamic service delivery, and charging based on usage. Beyond technological innovation, cultural adaptability in the organisation is essential for achieving an adequate degree of responsiveness.

Self-management of Virtual Organisations: Self-managed VOs are characterised by the ability to manage *at least* their operation and evolution without necessitating explicit intervention from the VO partners or other parties outside the VO. Self-management necessitates a form of integration that is enabled by the presence (or ability to form) an “autonomic” inter-organisational information system (IOIS) that supports the:

- *Negotiation and agreement of the conditions of involvement* of VO participants by means of electronic contracts whose operation is monitored and enforced by the IOIS;

- *Membership management and trust establishment* between the collaborating entities, be they the VO participants or the services and resources offered by the VO participants;

- *Specification, negotiation and distribution of policies* that control the sharing and aggregation of services and resources of VO partners in compliance with their agreement. Such policies need to be enforceable by the IOIS, and allow for adaptation (in real time) in response to changes of the context of interactions;

- *Specification, distribution and enactment of business processes*, which orchestrate the aggregation of information, services and resources in accordance to the consolidation of agreements between VO partners, VO-wide policies, and local (organisational) policies.

- *Resolution of conflicts or adaptation of VO operation* in response to alleviating the impact of violations of agreement or conflicts between policies, agreements, and business processes – both on the basis of their static description and their enactment.

The efficiency and effectiveness of self-management also depends on the extent that a VO achieves security and integration and in particular organisational transparency, shared leadership, and separability in VO operation and management.

Scalable Virtual Organisations: Scalability refers to the ability of the VO framework to be realised in different scales depending on its objective and the kind of the parties involved (e.g. large corporations, SMEs, solitary entrepreneurs). Scalability also allows the multiple interdependent layers of outsourcing as manifested by the constitution of recursive VO structures where similar VO formations may appear in micro- and macro- levels. This can be the case for example of a federation of providers offering high-performance end-to-end aggregate processes, where some activities are realised by aggregations of services provided by smaller-scale nested VOs and enacted over virtual execution environments which are also understood as VOs of execution hosts that federate resources in order to realise the enactment a service component.

Integrated Virtual Organisations: This characterises VO structures that comply with the typology of “dynamic networks”, as explained above, while they also support at least the following extreme aspects of an organisational network:

- *Organisational transparency* – meaning that although frequent and fine-grained interactions with customers are supported in order to facilitate mass-customisation, the co-operation of VO partners may not be visible to customers.

- *Cost-effective inter-organisational information system* (IOIS). A cost-effective IOIS requires the virtualisation of information and computation services and resources (at different levels of granularity) and their “just-in-time” integration across the boundaries of the VO partners. The formation of such a IOIS to accommodate the secure enactment of a collaborative business process within the VO should ideally take place at the time of demand. Its operation should respect the agreements between VO partners, the policies of the services and resource providers (who may maintain overall governance of the assets

the provide to the VO) and the service and resource distribution should be transparent to the consumer both within and outside the VO;

- *Shared control* – meaning that while every partner contributes to the operational management of the VO, it does not automatically control the whole VO, although it effectively maintains high-level control within its own local administrative domain;

- *Shared leadership* – meaning that while every partner maintains control of their own assets and serves their own interests, these *must* relate to, and *may* be partly overlapping with, the interests of the collective.

- *Shared access to resources and services* – VO member may consume shared resources or services of other members for the purposes of enacting a collaborative task;

- *Shared loyalty* – entities within a VO constituency contribute to the common objective of the VO but also serve the specific objectives of their own organisation.

- *Mission overlap & co-destiny* – there may be partners that are also doing business outside of the context of the VO (and have a partial mission overlap) in addition to those (having a complete mission overlap) that all business is conducted within the VO context. In either case, the mutual dependencies due to the sharing or resources, services and knowledge and the shared risks make the VO partners also more dependent on each other, therefore necessitating collaborations based on a sufficient level of trust;

- *Separability* – meaning that VO management is characterised by clear distinctions between specification and deployment, and between a VO-wide "global" strategic management and a "local" operational management.

To achieve the desired economical performance, such "integrated" VOs require a functional efficient corporative network. VOs may be embedded in a larger network of corporations, from which certain members are recruited to deliver the required performances.

4.2.2 VO Management concepts and innovative, added-value functionality

Current Web Services and Grid based VO Management solutions support only the VO membership function alone in most cases – listing VO members who are entitled to use VO resources. These solutions do not support the management of the risks associated with VO membership through:

- The identification of potential VO partners through reputation management
- The roles defined in business process models that VO partners perform to limit resource access and reputation transfer
- The contractual or SLA obligations on the VO for security and privacy
- The enforcement of policies derived from contracts for quality and timeliness of business process enactment

The Trustcom VO management component will not only provide a membership function but will also provide life cycle and context management functions to provide a generic VO

management layer that not only records membership, but also addresses the management of the risks of VO membership.

In addition, the TrustCoM VO Management solution will use declarative specifications of VO management processes, where appropriate. This approach will allow to publish and access commonly understood and approved protocols of VO Management and VO Members' interactions. Having these interactions explicitly choreographed will reduce the complexity of end-point software components of VO Members' software, thus increasing robustness, maintainability and efficiency of VO Management procedures. TrustCoM is using WS-CDL specification (a candidate W3C standard) to specify choreographies and emerging open-source tools, which allow modelling and validation of the protocols as well as end-point generation for WSDL based and BPEL-enabled services.

Tools for managing the identified risks are being developed in the Trustcom framework in the trust, BPM, contracts and policy components respectively. Putting them together to manage the risks of joining and operating within a VO is the innovation for VO management in Trustcom. The Trustcom VO management component is the generic VO application layer that ties together these Trustcom components. In practice, an individual VO will also incorporate specific application on top of the generic VO layer (e.g. aerospace modelling tools, remote learning tools) but it is the generic VO management layer that ties the other Trustcom components together, on which these VO specific applications can build.

By defining the standard VO Management protocols based on open standards TrustCoM contributes to establishment of VO Management patterns which may be useful beyond the scope of the project itself. VO Management subsystem should provide the services necessary for maintaining the VO structures, monitoring members' performance, enforcing VO policies, assigning members to play certain roles and perform task etc.

Each of the above mentioned VOM value-added features present a set of certain problems in the context of the dynamic VO conceptual model on one hand and the chosen technology (Web Services) on the other hand. One of the aims of TrustCoM VO Management research effort is to identify and capture the most common issues, which can take place in many different cases, and generalise these issues in order to provide solutions, which can be applicable to a whole class of VOM-related situations.

Naturally, this type of expertise cannot be “invented” or fully designed upfront, it takes time and careful refactoring of the initial design decisions until the solution to a problem reaches adequate level of wider applicability. However, following the approach of interoperability profiles (and that of patterns), we aim from the very beginning to single out the most common issues and put them in te context where we look for suitable solutions.

The existing Web Services Profiles (such as WS-I Basic/Security Profile of IBM B2B Basic Profile) address more technical level of interoperability, whereas TrustCoM aims to provide

related to aircraft design which contains specific relationships related to the analysis and design needs and the storage provision. We advocate that strict encapsulation be used for the nesting of relationships in order to achieve separation of context and be able to use relationships as unitary consistent groups.

VOs of any reasonable size will have large numbers of policies (both obligation policies and access control policies). Often policies do not exist in isolation but form consistent and coherent groups. For example, obligation policies must be accompanied by the authorisations permitting the actions specified in the obligation to be executed. Conversely, obligation policies may specify how consecutive unauthorised action requests may be handled or how access control policies should change with substantial variations in trust of the accessing parties or risk of the action being performed. These policies must co-exist within the system and therefore need to be deployed, enabled or disabled simultaneously within the VO. As a consequence we advocate the use of relationships not only as a grouping construct but also as a unit of deployment and instantiation. Administrators and VO managers can specify, deploy, enable and disable relationships but must do so on the relationship as a whole and should not manipulate policies individually. Relationships may contain other relationships as discussed above and may specify policies regarding the instantiation and deployment of their nested relationships.

Previous work has investigated enforcement of obligation policies by numerous enforcement agents distributed around the system[13]. This model has had mixed success as it has two main shortcomings: first it assumes that these agents exist, are instantiated and deployed in the system (in essence considering that a large number of objects in the system embed an event-condition-action rule engine), second it complicates the semantics and understanding of policy specification because although both obligations and access control policies refer to “subjects” the semantics of “subject” in the two types of policy is different. This has been the source of confusion and has complicated unnecessarily previous models. The TrustCoM framework advocates the use of the relationship objects themselves as enforcement point for the obligation policies. This enables a much simpler deployment and enforcement model for obligation policies without loss of generality. The relationship objects are stored and interpreted at the policy services (see D9) Note that a nested relationships in a parent relationship may be deployed and enforced at a different policy service than the one where its parent resides.

In VO environments there is a need to be able to re-use policy (relationship) specifications in different contexts. For example the relationships relating to storage provision will broadly entail the same policies in all the cases where it is used. Hence, we can achieve re-use of specifications by defining relationship types paramterised in terms of the roles of the various participants or characteristics of the policies concerned.

In the VO life-cycle policies are specified during both the VO formation phase as part of the GVOA but also throughout the operation phase as the VO needs to adapt to changes in user requirements and in the context in which the VO operates.

[13] N. Dulay, E. Lupu, M. Sloman and N. Damianou. A Policy Deployment Model for the Ponder Language. IFIP/IEEE Symposium on Integrated Network Management, Seattle, USA, 2001, IEEE Press.

Although it is theoretically possible to encode all adaptation actions as policies in the beginning, it is unrealistic to think that all requirements for adaptation can be anticipated during the formation phase. Therefore, new policies will need to be defined and instantiated at run-time during the operation of the VO. A user interface must therefore be provided to the VO administrator to achieve this which will take the form of a policy specification and command language. In the next section we will use examples of a pseudo-policy language in order to describe the concepts and their operation.

4.6.3 The Policy Specification Language (2Ponder)

The most appropriate form of interaction for an administrator is through a policy specification and command language. Although a graphical user interface could be conceived to fulfil the same purpose our experience has shown that graphical interfaces for policy specification are cumbersome, and do not easily allow the visualisation of several policies at the same time. Thus, the ability of a human user to read and assimilate a several policies at the same time is hindered by the idiosyncrasies of the user interface and does not lend itself to large specifications.

In this section we attempt to illustrate the concepts and operation of the policy sub-system from the view point of the VO administrator through the language constructs. **Note** that this is by no means the final or the concrete version of the policy language syntax and serves here only as a means of example for illustrating the concepts.

Let us first attempt to formulate an authorisation policy that permits project managers from Airco to start simulations and retrieve results from SciTech's simulation servers when the latter are active. It is likely that project managers and simulation servers will be respectively the subjects and targets of other policies so we define their specification separately:

```
subject     projman() -> role = "Project Manager" and role.issuer = "AirCo"

target            analysissrv(design) -> design.get("analysis_services/*)
```

The two lines above require further explanation. Subjects are used only in access control policies and denote the entities that are permitted to perform the operations. They are characterised by their attributes (e.g., role in a VO participant organisation) which are sent in credentials together with the request. The subject specification is therefore a condition over the attributes presented with the request.

Targets on the other hand determine the services whose operations are permitted to be invoked. Fundamentally, a target is therefore a set of services which must be registered in a domain structure or provided by the role. In this example analysis services are defined as all members of the "analysis_services" domain in the design role.

The authorisation can now be written as:

```
auth+ projman() -> (t = analysissrv(/roles/Scitech)).start()
                        -> t.sentResults("airCo")
```

```
                    ?  t.status = "active"
```

Similarly, we can define a new policy that allows project managers or any entity trusted above a certain threshold by AirCo to add new parameters to the simulation.

```
subject simdes() -> projman() or   (trust = "high" and

                                   trust.issuer = "AirCo")

auth+ simdes -> analysissrv(roles/SciTech).submit()
```

and an obligation policy that re-schedules the analysis calculations when new parameters have been added:

```
oblig paramschange(sim) -> sim.server.reschedule()
```

The obligation policy is interpreted by the relationship object itself which subscribes to the notification service in order to receive the events and performs the specified actions. This is described in more detail in the section below and in D9.

These policies can be grouped in a relationship or even better in a relationship type that can be instantiatied multiple times when new partners or application tasks are added to the VO:

```
type design_outsourcing(aerospace, design)

        subject    projman()   ->   role   =   "Project   Manager"   and
role.issuer = "AirCo"

        target          analysissrv(design)                        ->
design.get("analysis_services/*)

        subject simdes() -> projman() or    (trust = "high" and

                            trust.issuer = "AirCo")
        auth+ projman() -> (t = analysissrv(design).start()
                            -> t.sentResults("airCo")

                            ?  t.status = "active"

        auth+ simdes -> analysissrv(design).submit()


        oblig paramschange(sim) -> sim.server.reschedule()
```

This relationship type takes the names of the roles which participants are expected to play in the relationship as parameters and defines the policies that are part of that relationship. Policies are expressed in terms of the role parameters.

Whthin a specific context this relationship type can be instantiated with specific parameters e.g.,

```
inst airco_scitech = design_outsourcing(airco, scitech)
```

By default the instantiation occurs within the policy server where the instantiation command was executed. However, it should also be possible to instantiate a relationship on a different server e.g.,

```
inst airco_scitec@policy_server_1 = design_outsourcing(airco, scitech)
```

The relationship for the storage provision can also be specified in the same way. This specification also shows obligation policies triggered by reputation services.

```
type storage_provision(design, mainstorage, backupstorage)

            subject designers() -> role = "Designer" and role.issuer =
"SciTech"

            target            storagesrv() -> mainstorage.get("storesrv")

                                                                    +
backupstorage.get("storesrv)"
            auth+  designers() -> storagesrv().read()

                                                        ->
storagesrv().write(x) ? size(x) < 40
      auth-             designers() -> storagesrv().start()

                                                        ->
storagesrv().stop()
            event       quotaexceeded() ->

                                                              e             =
/notifications/topics/storage/quotaexceeded

                                                              ? e.excess < 10

            event       untrustedsubject() ->

                                                              e             =
/notifications/topics/trust/reputationchange

                                                              ? e.value   < 50
and e.actor = backupstorage
            oblig       e = untrustedsubject()

                                                        ->
/services/membership.startremoval(e.actor)
```

The relationship corresponding to the overall VO can then be written as below. This defines a new relationship type which imports the other two type definitions and creates instances of it on specific policy services.

```
type aird(aerospace, design, mainstorage, backupstorage, infrastructure)

            import design_outsourcing
```

```
            import storageprovision

            inst airco_scitech@infrastructure/policysvc1 =


      design_outsourcing(aerospace, design)

            airco_scitech.enable()

            inst scitech_storage@infrastructure/policysvc2 =


      storage_provision(design, mainstorage,backupstorage)

            scitech_storage.enable()
```

The instantiation of this relationship type would lead to the instantiation of the entire sub-structure of the VO described in the figure.

4.6.4 The enforcement model

This section provides an overview of the enforcement model of both obligation and access control policies. It focusses on the same scenario as above and the diagrams illustrate the operations occurring in the policy sub-system during policy enforcement. More details of the architecture providing this enforcement are given in D9 and the conceptual model underpinning the abstractions is described in D16.

4.4.6.1 Relationships

Relationships are instantiated and enabled in policy services. A composite relationship that may contain other relationships may direct different relationships to be instantiated on different policy services. This model allows arbitrary distribution of policies encapsulated in relationship objects across any number of policy services that may be distributed across the VO. Depending upon the VO model e.g., CE or AS scenarios different distribution models may be suitable. Small VOs may rely on a single service whereas larg-scale VOs may choose to have policy services distributed across all partner sites in the VO. The ability to define relationship types provides re-use of the specifications and allows the instantiation of multiple instances of the same relationship type in the same VO. This permits to cater for VOs that allow competitors i.e., multiple providers of the same type of services. Figure *6* below shows the relationship between the policy services in the example used and how policy services collaborate between them.

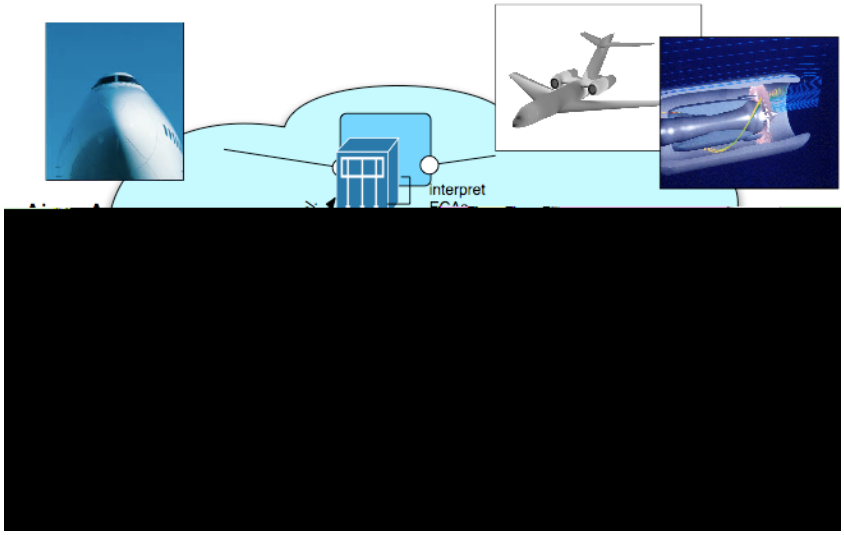

Figure 6 Distribution of relationships across policy services.

4.4.6.2 Obligation Policies (ECA model)

The obligation policies define which management or corrective actions need to be performed in response to events happening in the system. Once instantiated and enabled in a policy service the relationship object verifies the event expressions included in all its obligation policies and subscribes to the notification service in order to receive those events. When the relationship is enabled, each relevant event received leads to the conditions of the policy being evaluated. If these conditions are satisfied the management actions specified as part of the policy are performed on the target services (see figure below). Note that target services may be either:

- application services which participate in the business process of the VO
- adminstrative services such as the reputation service and the membership management services of the VO
- infrastructure services such as the notification services or trust and security services of the VO.

Figure 7 Enforcement of obligation policies (ECAs)

relations can only be considered in restricted cases where the precedence relations are simple and well defined. Conflict resolution through complex precedence relations greatly affects the usability of the policy system since it becomes impossible for a human person to predict which policies will apply and which will be ignored through other policies' precedence over them.

Broadly, two classes of conflicts can be distinguished: conflicts which stem directly from the policy specification and application-specific conflicts for which additional specifications are needed in order to characterise the conflict. The well-known separation of duties conflicts fall within the latter category.

Policy Validation refers to the ability to determine whether a set of policies can be implemented in the existing system in which they are to be deployed. Here again we can distinguish two types of validation: static validation where policies are considered against the other's component's specifications and static capabilities e.g., the interface of a web-server and the maximum number of connections it can support, or dynamic validation where policies are considered against the current state of the system in which the policies are to be deployed. The latter also takes into account current load on servers, context and other dynamic parameters that can be determined only at run-time.

Policy Refinement refers to the ability to translate policies from a higher level of abstraction into a more concrete specification. In general this cannot be automated and requires human intervention. Through innovative methodologies and by constraining the problem to specific application domains it is however possible to automate parts of this procedure[14].

TrustCoM does not aim to develop new techniques for policy analysis and refinement. Instead we will look at extending and adapting if necessary techniques for off-line policy analysis developed by partners in other projects. Also it is worth noting that existing implementations of XACML-based policy decision points do not easily handle negative authorisations (i.e. prohibitions) thus reducing the potential for conflicts in the specification. The run-time handling of conflicts and run-time conflict resolution is a research topic in its own right which existing work rarely if at all attempts to address. It is not foreseen that TrustCoM will be able to address this issue.

4.6.6 Standards and Profiling

The role of standards and in particular the WS-* series of standards remains an important aspect of the TrustCoM framework. The main elements influenced by standards in the policy sub-system are the authorisation PDP and its interfaces with both external and internal components. This is because authorisation policies and protocols for interacting with PDPs have been in the standardisation arena for several years and are reasonably well understood even if consensus is sometimes more difficult to reach. In our case the authorisation PDP will accept as input

[14] A. Bandara, E. Lupu, J. Moffett, A. Russo. A Goal-based Approach to Policy Refinement. IEEE Workshop on Policies for Distributed Systems and Networks, 5th IEEE Workshop on Policies for Distributed Systems and Networks, (Policy 2004), New York, June 2004.

XACML policies (potentially with extensions for delegation of authority) and its interface to the PEP will be through the use of XACML request embedded in SOAP messages. We are also considering in future versions the SAML 2.0 profile for XACML for this interface. There seems to be no current adequate protocol for the policy service/PDP interface although some of the standardisation bodies such as GGF seem to show interest in this. The closest relative for such a protocol from the standards area would be COPS used in provisioning mode but this is a protocol aimed at network management and within the context of TrustCoM it would provide little benefit.

The specification deployment and enforcement of obligation policies in the form of event-condition-action rules has not been the subject of standardisation. XACML has the notion of obligation policies but their scope is restricted to the enforcement of access control decisions in the PDP. The most relevant standard would be DMTF's PCIM with associated extensions but the specification notation and encoding are rather cumbersome and assume the use of a CIM model which does not exist for many of the elements in the TrusCoM framework. However, it is our intention that the event specification and subscription mechanisms will follow WS-Notification.

Although WS-Policy and WS-SecurityPolicy could in a general sense be used to encode policies in an XML form, WS-SecurityPolicy is really aimed at configuration of Web-service security mechanisms and WS-Policy only provides a general encoding without specific policy elements or semantics.

4.6.7 Conclusions

The framework for the policy sub-system described above aims to provide a flexible and convenient means of specifying deploying and implementing adaptation and access control policies in the TrustCoM framework. Adaptation is primarily realised through obligation policies in the form of event-condition-action rules which are enforced at policy-services that can be distributed across the VO. Access control is realised through authorisation and delegation policies that are deployed to authorisation policy decision points in an extended XACML format and enforced by the PDP upon evaluating requests received from the enforcement points. Both obligation and access control policies are grouped in relationships which provide a unit of deployment, instantiation and re-use for policy specifications in the TrustCoM framework. Relationships may be nested and their implementation can be distributed across policy-services. Thus relationships provide a structuring principle for policies within the TrustCoM framework. Details of the conceptual model and architecture for the policy sub-system are given in D9 and D16.

EN/VO Infrastructure - HLRS

The Infrastructure related technology specifications are mainly basing on the WSRF-related set of specifications, namely WS-ResourceLifetime, WS-ResourceProperties, WS-BaseNotification, WS-Topics and, up to some degree,

WSDM. As opposed to the more WS-Transfer related set, i.e. WS-Transfer, WS-Eventing, WS-Enumeration and WS-Management.

The choice mainly bases on the fact that at the time implementation started, public interest, as well as available tools support seemed to indicate a higher interest in WSRF. However, it seemed sensible to not rely on this one specification, as the advantages and disadvantages of WSRF and WS-Transfer seemed about equal – recent research by the University of Virginia[15] seems to support this view. We therefore chose an approach of "adapting" the WSRF specifications in a way that would allow comparatively easy code movement to the set of WS-Transfer related technologies, if so required. This shall be examined further in the following:

4.6.8 WS Resources

The main goal of the WSRF (WS-ResourceLifetime, WS-ResourceProperties) and WS-Transfer specifications consists in providing standard web service support for "statefulness", i.e. providing some standardised interaction between a customer and a service that offers storage in the form of state. The main difference between the two specifications consists in what the developer considered necessary for state support:

According to the WSRF approach, resources are "organised" as sets of properties that may be accessed individually, where a property may be anything from individual parameters to a serialised dataset. Due to limitations in WSDL, properties are not dynamic, i.e. even though the structure of their contents is flexible, it is not possible to add or remove initially declared properties – thus a service that wants to expose further resource parameters will have to be restructured and re-instantiated.

As opposed to this, WS-Transfer does not support the notion of individual properties and only allows for retrieving and setting all status-related parameters *as a whole*, i.e. as a serialised XML document. This way, the structure is completely dynamic, even during run-time, but parameters can not be queried individually and passing the whole document may cause delays due to bandwidth restrictions.

In order to harmonise the two specifications and to keep changes to the code minimal when swapping from WSRF to WS-Transfer, we do not expose status parameters individually, but offer the whole state-related information via one method (respectively one resource-property) only, that will allow for accessing the status *as a whole.* This way, the schema of the specification remains unaltered, whilst the code is prepared to allow for the "simpler" approach of WS-Transfer.

[15] Marty Humphrey, Glenn Wasson, Yuliyan Kiryakov, Sang-Min Park, David Del Vecchio, Norm Beekwilder: *Alternative Software Stacks for OGSA-based Grids.* University of Virginia Department of Computer Science Tech Report CS-2005-06, April 2005.

4.6.9 Manageability

A similar approach will be taken with respect to manageability issues, where the opposing specifications are WSDM and WS-Management for the WSRF and the WS-Transfer domain respectively.

4.6.10 Notifications

WS-BaseNotification versus WS-Eventing alter only slightly with respect to functionalities and may hence be used alternately without too much implementation overhead by "simply" changing the used schema. The main difference, however, consists in the respective usage of topics-information: whilst WS-BaseNotification provides its own specification regarding this, namely WS-Topics, WS-Eventing relies on a simple filter-like approach that may be as basic as comparing strings.

Again, the TrustCoM framework tries to cater for easy code movement and will thus try to identify a common ground. The simplest approach with respect to this would consist in serialising the WS-Topics compliant topics into a string which may be used as WS-Eventing topic. This will be further supported by not making use of the additional WS-Topics functionalities, like topic-spaces (cf. WS-Topics[16])

4.6.11 Additional issues

Besides for the WSRF and/or WS-Transfer related technologies, the infrastructure also makes use of the following specifications:

- WS-Addressing

  which will generally be used straight-forward, i.e. to carry the unique id of the service as part of an Endpoint Reference. The same mechanism will be used to extend the EPR with respect to additional context-information, like the unique id of a SLA valid for this provider-customer relationship etc. just as intended by the WS-Addressing specification.

- UDDI

  the exact usage details regarding the UDDI specifications are as yet, at the time of writing this, not entirely certain. It is however most likely that other discovery-related repositories will make use of their own specifications rather than adapting UDDI, in which case UDDI will be used without any alteration as a repository-technology for storing the WSDLs of the available services.

- WSDL

  which is extended according to the definitions in WSRF[17] and WSDM[18]. Note that we use document-literal encoding due to compatibility issues.

[16] http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.pdf

[17] http://www.globus.org/wsrf/

[18] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm

In summary, the EN/VO Infrastructure does not (yet) extend the above listed specifications, hence we did not replicate the profiles here, as they can be taken from the respective sources.

5 Future Plans

The present state of the Trustcom framework is only version 1 of three planned versions which will be released at six monthly intervals. This document is consequently only a step towards the completed Framework. Some sections presented here (e.g. SLA) have defined the level of detail that is expected of the profile part of the framework, while others (e.g. VOM) are still in a less developed form.

It is planned to progress all components to the full level of profile descriptions.