Deliverable
D63

TrustCoM Framework V4

AL1 – TrustCoM Framework

Michael D. Wilson, CCLRC
Alvaro Arenas, CCLRC
Lutz Schubert, HLRS

06/03/2007

V3.9

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

Networked business and governments

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 Science AB,
Europaeisches Microsoft Innovations Center GMBH,
Eidgenoessische Technische Hochschule 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 Kent,
International Business Machines Belgium SA .

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: AL1

Activity: 1.2

Work Package: WP27

Task:

Document title: **TrustCoM Framework V4**

Version:	**v3.9**
Document reference:	
Official delivery date:	**31/01/2007**
Actual publication date:	**06/02/2007**
File name:	D63 TrustCoM Framework V4.doc
Type of document:	Report
Nature:	official deliverable

Authors:	CCLRC: Michael D. Wilson, Alvaro Arenas; IBM: Jakka Sairamesh; SAP: Florian Kerschbaum, Philipp Robinson; UoK: David Chadwick; NRCCL: Dana Cojocarasu, Tobias Mahler; EMIC: Christian Geuer-Pollmann, Joris Claessens; HLRS: Lutz Schubert; BAE: David Golby; ATOS: Ignacio Soler; SICS: Pablo Giambiagi, Thomas Olsson, Andres Martinelli; ETH: Jürgen Doser
Reviewers:	Christian Geuer-Pollmann; Bernhard Thurm, Dana Cojocarasu
Approved by:	

Version	Date	Comments
V3.1	02/11/2006	Updated document structure
V3.1.1	08/12/2006	CCLRC: updated Introduction

V3.2	02/01/2007	Updated section II.1, added requirements summary
V3.3	10/01/2007	Added business model discussion to section V.1
V3.4		
- V3.7	till 16/01/2007	Updated relationship details in section III
- V3.9	till 19/01/2007	Added methodology sections per subsystem (chapter VI)
V3.10	6/3/07 mdw	Cleaned up to remove partner names from sections

Table of Content

Introduction

The TrustCoM project[1] has developed this Framework for Trust, Security and Contract management for dynamic virtual organisations operated through an open service architecture. The Framework includes a way of conceptualising the Trust, Security and Contract issues associated with dynamic virtual organisations, an architecture in which the operating open services can be implemented and profiles of proposed and standardised web service specifications tailored to VO application. The project has also produced a reference implementation of the architecture, and demonstrators whose evaluation shows the strengths and weaknesses of the framework.

VO Management as developed for academic Grids, has previously only addressed membership issues and has so far ignored the Trust, Security and Contract management issues addressed in the TrustCoM framework. Consequently, these aspects of the Framework and the architecture are innovations. Supply chain management systems for large organisations do address contract development issues but do not address most of the security and automated contract and SLA monitoring and policy enforcement issues that are addressed by TrustCoM, so these remain innovations. Stand alone systems address role based security, but not the contractual context that TrustCoM does, so even the approach to this technology is innovative. The main innovations in the TrustCoM approach are in integrating these technologies to refine a contract and ancillary SLAs stated in business terms into deployable processes that monitor and enforce those agreements between organisations.

The framework document provides an overview of the results achieved in realising the TrustCoM framework.

This document describes the final results of the framework design process in TrustCoM and as such provides an update to the Framework version 3 [3]. Like version 3, it spans the originally three deliverables of the framework, namely the concepts, architecture and profile specification. Besides further refinements and alignment with the ongoing implementation work within TrustCoM, this cycle provides in particular the following updates:

- enhanced conceptual architecture

  Chapter II of this document now addresses the link between conceptual requirements (chapter I) more explicitly, so as to make the rationale behind the architectural choices in TrustCoM more understandable. Accordingly, the *purpose* of the selections and hence the *usability* becomes more obvious.

  In particular, chapter II now contains an explicit list of requirements and how they were addressed in the TrustCoM architecture.

[1] The project is structured into Action Lines, Activities and Work Package which produce internal deliverables as well as externally available ones such as this. Consequently, references occur in this document to these internal project management entities which are meaningful to those within the project, but may not be to other readers. The authors and editor ask you to tolerate these references.

- updated relationship models

With regards to the latest development and conceptual discussions, in particular from the non-technical partners, the relationship models have been updated to provide a more accurate and both technical and non-technical feasible view on the *usage* of the framework.

- business oriented deployment model

As opposed to version 3 of this document, the deployment model (chapter IV) does not discuss detailed distribution of components across different infrastructures any longer as this was restricting the actual capabilities of the Service Oriented Architecture approach.

The section now provides general deployment recommendations and addresses business model (section I.2) specific issues with respect to deploying the TrustCoM framework.

- an assessment of the testbeds from the business model perspective

The testbed scenario descriptions (chapter V) now provide an additional section wherein the scenarios are analysed with respect to the business models (section I.2).

- a methodological section (from a conceptual stance)

Chapter VI of this document provides an overview over the most relevant aspects to consider when addressing an *implementation* of the framework. This covers in particular “information sets”, i.e. what kind of information is required for the functionality of the respective components and in what order does it need to be provided.

Note that this section strongly relates to the Profiles section of previous versions of this document which now has moved to Appendix A.

The structure of this document follows the initial structure by first providing an insight into the underlying ideas and concepts that guide the development process. It also specifies the requirements that define the basic structure of the framework. Section I of the document presents the conceptual basis of the project, ranging from the fundamental concepts in TrustCoM (trust, contracts, security) to the underlying business model.

The following sections (II & III) then provide an overview over the architecture aiming at fulfilling these concepts as envisaged by the TrustCoM consortium. Since one of the main issues of a service oriented architecture as pursued by the project consists in realising a reconfigurable, “plug & play” infrastructure, the TrustCoM framework cannot be easily depicted using classical UML diagrams, as these do not allow for representing actual flexibility of the framework (see section II). Hence TrustCoM introduces the so-called “relationship view” on the architecture (section III) that represents the relationships between components rather than the actual deployment and interactions.

As such a view does not carry information about how to set up a TrustCoM framework, respectively which components are required under what circumstances, section IV provides deployment details covering the business models discussed in section I. As such this section addresses some implementation issues as they are faced by Action Line 2 of the project.

Since these views on the architecture - relationship and deployment - do not provide an easy to follow description of the actual transactions within a realised TrustCoM virtual organisation, section V exemplifies the architecture by means of two potential deployment structures based on the two testbed scenarios. Whilst these examples cannot cover the full complexity of the Scenarios, they exemplify how the TrustCoM architecture can be used to address various business requirements.

Many of these issues have been addressed by other project deliverables [5][6] and previous versions of this document [2][3] and we will refer to these documents rather than repeating all the information presented in them.

Finally, section VI elaborates the relationships between components by detailing the message requirements of the individual subsystems from a *conceptual* stance, i.e. examining the type of information required for providing the respective functionalities. This section provides a basis for any implementation task of TrustCoM specific components and is hence partially derived from the integration work of the reference implementation.

I The TrustCoM Conceptualisation

This section describes the concepts underlying the TrustCoM framework - these concepts have been identified by examining common business requirements, as e.g. represented by the two testbed stakeholders (BAE and ATOS ORIGIN) within the project. As such, this section depicts the main issues involved in realising Virtual Organisations and how they could be overcome. Notably, it does *not* describe all the goals to be fulfilled within the project's lifetime, but goes beyond it in scope: though the architecture tries to address most of the technical aspects, the concepts in particular cover the so-called non-technical issues in TrustCoM, like trust and legal relationships. Obviously, some of these aspects may never be realised on a pure technical basis (e.g. replacing a lawyer), but TrustCoM depicts how actual implementations and these aspects relate to each other and how the latter may be supported through usage of the TrustCoM framework, as e.g. with reputation: any technical realisation can only cover a part of the full philosophical discussion regarding trust and trustworthiness.

The description starts with the vision that TrustCoM is trying to achieve and which business model underlies this vision. It will then move onwards to detailing the aspects identified in separate chapters, e.g. contracts, trust and security.

I.1 The TrustCoM Vision

Given the economic competitiveness of a global economy, and the conflicting desire for a high quality of life in Europe, it was agreed by the heads of government at Lisbon in 2000 that Europe was entering a knowledge economy rather than one based on manufacturing or agriculture. In a knowledge economy, competitive advantage comes from the flexibility of organisations to respond to market opportunities. One mechanism to manage such flexibility efficiently is to automate the supply chain management for large organisations, or to provide environments to support the formation of Virtual Organisations (VO) of SME and large organisations which can recruit sufficient resources to take advantage of the opportunities where no organisation could alone.

Such an environment to support the formation and operation of VOs has to both be trusted itself, and provide a basis for trusting other organisations with which business could be done. Trust between VO members can be supported by each being transparently aware of the obligations and performance of others, so that business risks are both mitigated, and monitorable. The TrustCoM project pursues the goal of supporting the realisation of dynamic virtual organisations in a secure and contract managed environment. Thus TrustCoM envisages specific structures of collaboration between participants that actually form the basis for the framework as described in this document.

Business collaborations of the form envisaged by TrustCoM have significant impact on legal, business and technical resources of each participant in a VO. In particular, each participant needs to ensure the legal compliance of its interactions with other partners, the integrity of the business process within which it is involved, its reputation with regards to performance and service delivery, and the availability and confidentiality of its shared resources according to its agreement with the VO.

I.2 TrustCoM's Business Model

The TrustCoM framework defines a set of trusted services for enabling VO management in complex and ad-hoc business networks. The framework provides services that facilitate business collaborations in supply-chains and value-chains. For this framework to succeed over the Internet and in one or more industries a few comparative business models need to be considered, and were evaluated in this project. In general, collaborative applications in business environments can apply the practices developed by TrustCoM across standardised environments built on the Internet.

This section considers TrustCoM's two major business scenarios for the analysis and investigation of various feasible business models. In general TrustCoM VO management provides a contract based mechanism to handle VO lifecycle management. The VO management tools and methods apply to general collaborative networks of members. We consider business criteria and perspectives in order to evaluate the models based upon interaction, trust establishment, revenue generation and cost management. The following are the objectives for enabling business models analysis for TrustCoM:

- Recommend and evaluate business models based on cost, revenue and execution for the CE[2] and AC[3] (Ad-hoc Collaboration) business scenarios
- Identify and use critical business and technical criteria for evaluating the business models
- Compare and contrast the various business models based on revenue, cost, efficiency, and others
- Provide recommendations on enabling cost efficient interoperability, enforcement and monitoring requires standards and tight integration
- Recommend profitable TrustCoM models needed for social and business reasons

We consider multiple variations of business collaborations involving one-to-one and one-to-many contracts between collaborators with and without third-party trusted entities playing a role in reputation, evaluation and stability. The main business models are variations of the models described below.

- Models based on one-to-many contracts (VO managed by the initiator) which tend to have higher costs (e.g. transaction monitoring), complex interconnection, less flexibility and lower revenue when compared to third-party driven interoperation, integration and trust.
- Models based on third-parties hosting the VO which tend to have higher flexibility and better cost management in VO formation in the CE and AS scenarios. Third-parties focus on trust building based on reputation mechanisms, contracts and enforcement of contracts. The revenue models and costs are dependent on transaction volume, subscription rates and interoperation costs.

[2] Collaborative Engineering Scenario

[3] Ad-hoc Aggregated services Scenario

I.2.a Approach and Methodology

The following is the approach and methodology taken to evaluate business models in order to study methods of cost effective and streamlined management of Virtual Organizations:

- Defining the main criteria
- Define the business metrics for criteria
- Design and develop business models for VO management, CE and AS scenarios
- Compare the business models
  - Private versus public third-party
  - Private one-on-one interconnections
  - Buyer and seller controlled third-party
  - Investigate Game models for business interoperability
  - Evaluate the revenue and cost structure for each of the models

I.2.b Business Models for TrustCoM Scenario

In this section, we present the business models evaluation for TrustCoM VO management and processes. These models can be applied to CE and AS Scenarios combined for TrustCoM VO management

1) One-to-One and One-to-Many Models

In the Figure 1 below, we illustrate two models of interoperation and integration between the VOs and the VO initiator. The first model (1B) is a one-to-one interaction between the Enterprise (VO manager) and the partners. The partners for example include suppliers, dealers and other networked businesses. In Model 2B, the interaction is done through a Trusted Third-party with all the partners grouped into one consortium and managed as a single entity.

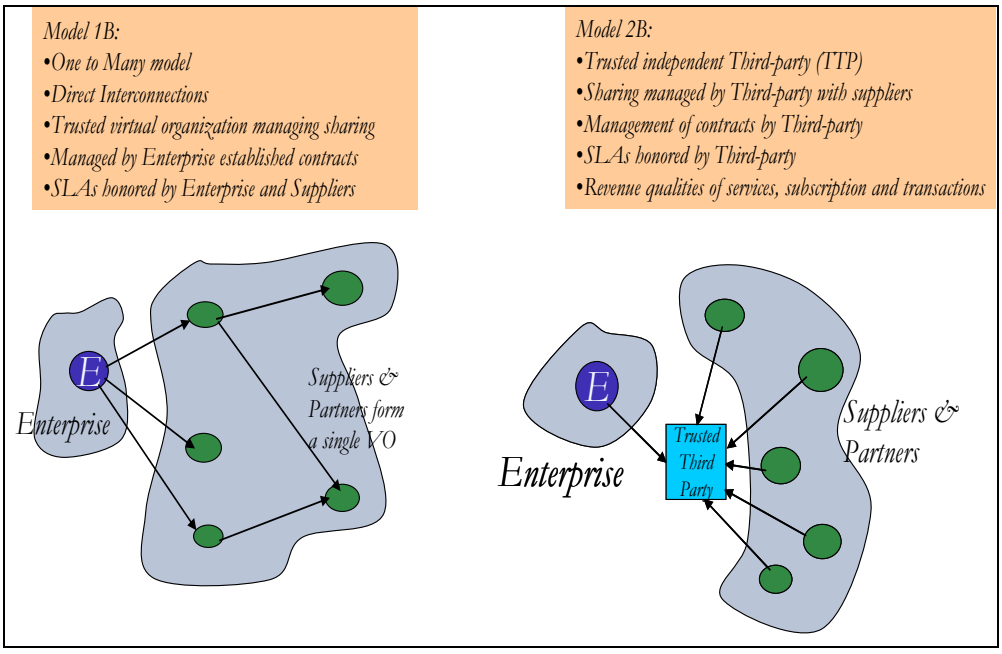


Figure 1: Models 1B and 2B

2) Trusted-Third-Party Consortia Models

In the Figure 2 below, we illustrate two models (3B and 4B) of interoperation and integration through a trusted third-party and multiple consortia. The third-party provides mechanisms for transactions, reputation, integration between multiple VO managers and trusted consortia. In model 3B, we consider a single VO manager (or initiator) and multiple partner consortia. The VO manager can have a stake in the trusted third-party to enable the transactions and interoperation. In model 4B, multiple VO managers interact with various partner consortia through the Trust-Third party.

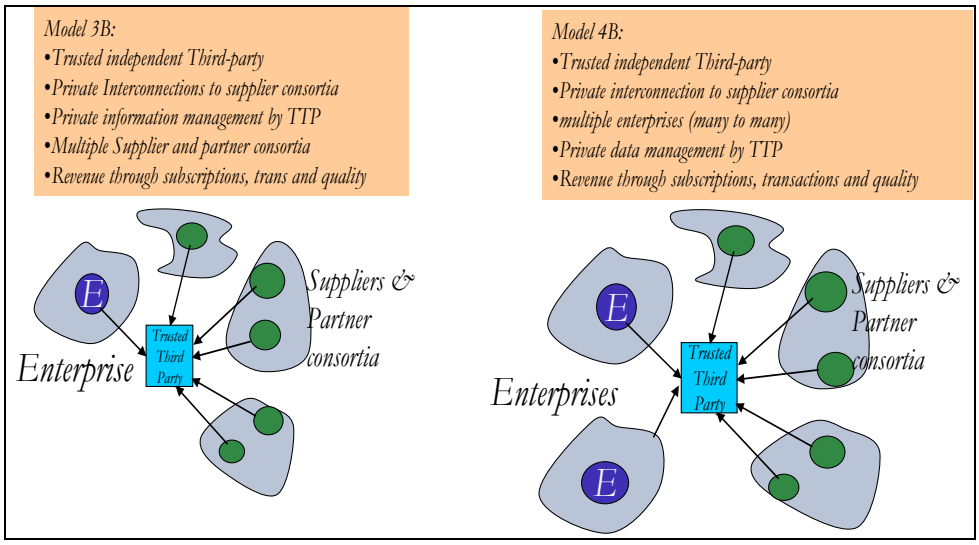


Figure 2: Models 3B and 4B

3) Partner Managed Consortia

In the Figure 3 below, we present two more models for TrustCoM scenarios. In the first model (5B) buyer consortia form and invest in a trusted third-party to manage the interactions with other partner virtual organizations. The second model (6B) considers supplier consortia that manage the trusted third-party for interaction. The models can be applied to VO management in general and to the CE/AS scenarios presented by TrustCoM. In general

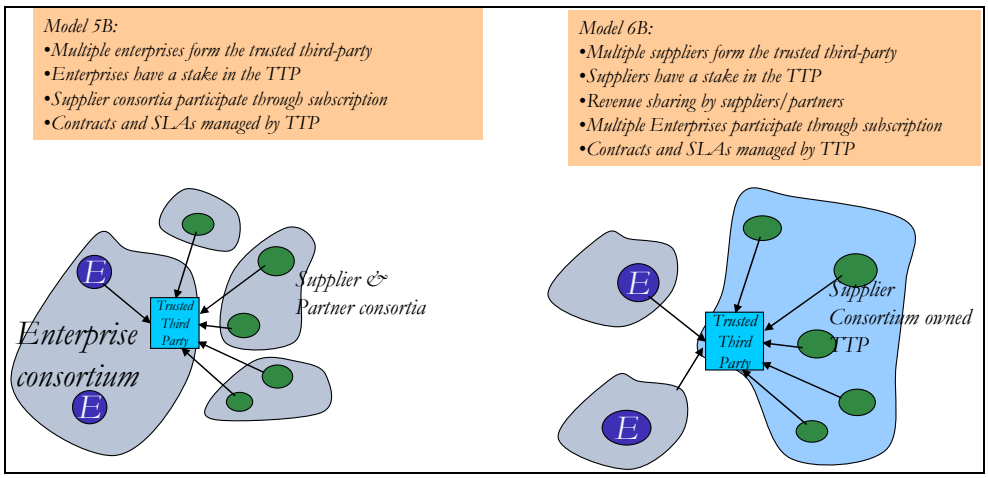


Figure 3: Buyer and Seller Managed Third-Party Models

I.2.c Comparison of the Models for the CE Scenario

Evaluation Criteria	Model 1B	Model 2B	Model 3B	Model 4B	Model 5B	Model 6B
Financial Revenue Costs Assets RoI and RoA	Subscription revenue Interconn and monitoring costs RoI and RoA driven by Enterprise efficiency	TTP Revenue based on Transaction, subscription and QoS Partner managed their own RoI/RoA	TTP revenue based on Trans, subscription and QoS Partner managed RoI and RoA	TTP revenue on trans, subscripion and QoS Partner managed RoI and RoA	TTP revenue on subscription. Enterprises have stake in TTP Partner managed RoI and RoA	TTP revenue on subscription. Suppliers and partners have stake in TTP Partner managed RoI and RoA
Organization	Enterprise managed Long contracts Supplier monitoring	TTP managed Contracts and SLAs through negotiation Monitoring by TTP	TTP managed Multiple consortia Multiple VOs Complex mgmt	Many-to-many TTP managed Multiple VOs Complex mgmt	Private Many-to-many TTP managed Multiple VOs Complex mgmt	Private Many-to-many TTP managed Multiple VOs Complex mgmt
Trust and Security	Enforced by Enterprise Enforced by partners Enforcement is complex	TTP enforcement of trust and security Monitoring done by TTP Enforcement is less complex	Enforced by TTP Enforced by partners Monitoring by TTP	Enforced by TTP Enforced by partners Monitoring by TTP	Enforced by TTP Enforced by partners Monitoring by TTP	Enforced by TTP Enforced by partners Monitoring by TTP
Technology	Contracts establishment SLA mgmt Web Services interconnection	Private information store, confidentiality Policy driven interconnection Web Services interconnection	Private information store, confidentiality Policy driven interconnection Web Services interconnection	Private information store, confidentiality Policy driven interconnection Web Services interconnection	Private information store, confidentiality Web Services interconnection	Private information store, confidentiality Policy driven interconnection Web Services interconnection
Business Process Metrics	Not efficient Substantial monitoring costs Enforcement costs High costs process management Semi-automation for processes is feasible	Processes managed by TTP Monitoring by TTP and partners High costs for process management Semi automation for processes is feasible	TTP managed processes High costs for process management and monitoring Semi automation for processes is feasible	TTP managed processes High costs for process management and monitoring Semi automation for processes is feasible	TTP managed processes High costs for process management and monitoring Semi automation for processes is feasible	TTP managed processes High costs for process management and monitoring Semi automation for processes is feasible

For TrustCoM to be lucrative and valuable a roadmap needs to be set to include current and future scenarios for evaluation and commercialization of assets or TrustCoM services to be applied to range of applications in business and social value chains.

I.3 TrustCoM's VO concept

TrustCoM has taken a very broad concept of a VO which includes:

- the shared resource VO consortium currently used in academic research grids;
- the business process role based collaboration hosted by a service provider which has become popular in the virtual enterprise research community to support SMEs (exemplified by the Ad-hoc Aggregated services Scenario);
- the supply chain partnership which is well established in existing businesses (exemplified by the Collaborative Engineering Scenario).

In line with the descriptions of [18], we distinguish four main lifecycle phases of a Virtual Organisation: (1) Identification and Discovery, (2) Formation, (3) Operation & Evolution and (4) Dissolution & Termination. In addition, we regard the initial actions to be performed by any Service Provider in order to enable it for integration into a Virtual Organisation such as envisaged by TrustCoM as an additional "phase", called "preparation phase" in the following.

The main tasks to be performed during individual operations for each phase are described here as scenarios.

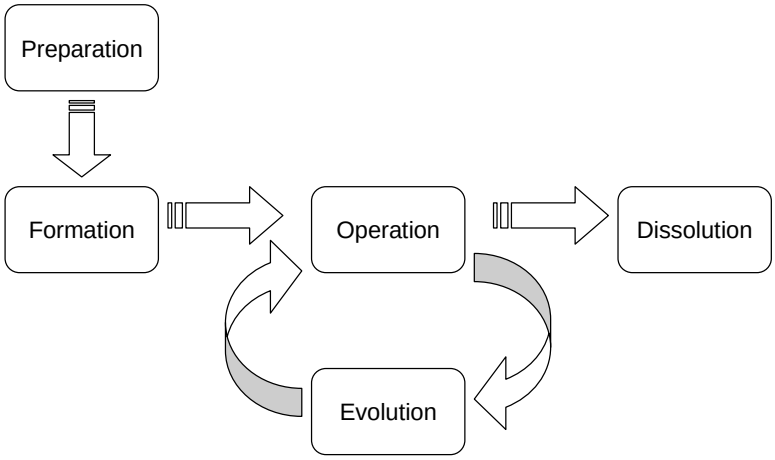


Figure 4: the lifecycle phases of a Virtual Organisation

1) Formation of an Enterprise Network (Preparation)

The first stage on VO formation is the formation of an Enterprise Network (EN) to provide a pool of organizations willing to join virtual organizations. Organisations must register with an EN register which acts like a yellow pages telephone book – listing the organization and the services that they are willing to provide. It is planned to take guidance from other IST projects that are investigating the issues of VO breeding environments in order to resolve this issue. The EN register and other EN and VO infrastructure services will be hosted by a provider. Business models are presented in other TrustCoM deliverables that show how the hosting of EN and VO services could be a profitable business in itself, probably as added value additions to basic ISP provision.

2) Establishment of a Virtual Organisation (Identification and Formation)

An organization which is registered as an EN member identifies a business opportunity and has the intention of creating a VO to meet it. They become the VO initiator, defining the goal of the VO, and try to discover the organizations required to make up the VO to achieve the business objective. The VO initiator will interact with a service provided by the hosts of the EN and VO infrastructure to guide him through the creation of a VO – the VO Management service, which is one of several VO services that will be introduced in the scenario.

Given a specific business objective (provided by a customer or by the VO Management organisation itself) to be realized by a virtual organisation, the VO Management service triggers the derivation of a business processes according to a collaboration definition by contacting BP Enactment. The latter now queries (known) BP Template Repositories for collaboration definition that realises the given task. Such templates contain the next

highest-level description of activities, information that is related to the roles involved in realising the processes (i.e. descriptions of the services that fulfil the individual tasks), coordination information (how the services have to interact) etc. which are passed to VO Management for partner selection. It is an open issue to be addressed by the evaluation of the TrustCoM demonstrators whether the level of description of the business objective and the collaboration definition are defined at an appropriate abstraction for the available definition of the market opportunity and the envisaged structure of the VO at this stage of the process. They may be either too abstract or too concrete, in either case the mismatch between the representations offered by the VO management services and the conceptualisation of the human VO initiator will increase the risks of the VO failing - even at this early stage,

VO Membership Manager is invoked with the collaboration definition's role related data containing information[4] about the structure of the service (operations, interface etc.), the quality it has to fulfil (SLA) and its trustworthiness[5]. This information is passed to the Discovery Service, which for each role to be filled, contacts a set of (known) repositories and returns a (sorted) list of potential organisations that meet these requirements.

Once the Membership Manager has received this list of potential participants, the SLA Negotiator on VO Management side is triggered to negotiate the actual terms with the Application Service Providers (starting with the most suitable ones), until all roles are manned. If negotiation fails to cast a specific role, i.e. if none of the respective organisations meet all requirements, the business process to be executed by that organisation and its requirements need re-evaluating.

As soon as all participants in the virtual organisation have been identified, VO Management triggers distribution of the relevant information to each of the VO members – this includes:

a) required credentials to access other members,

b) interaction and coordination information, like what data to pass when between services

c) VO agreements and policies, as well as

d) other configuration data (contact information, notification topics etc.).

Once all participants have confirmed their configuration, the VO manager is ready to instantiate the VO and enact the overall collaboration definition.

3) Normal operational work

With all VO members configured, BP Enactment starts the execution of the overall collaboration definition by triggering the first Application Service Provider(s) of the workflow and forwarding relevant execution data to it (like input values or location of data files).

Generally, the Application Service itself is responsible for triggering the execution of follow up tasks by forwarding its output data to the Application Service Provider(s) next in the overall collaboration definition (the relevant information, like which services to contact and

[4] Further information types may be added in the course of the project

[5] Note that „trustworthiness" as used in TrustCoM relates to „reputation" of the respective service provider. Accordingly, services that have not yet gained such a reputation need particular treatment.

which data to pass, has been provided during VO formation for each derived BP per role – cf. section I.4).

At checkpoints in the enacted business processes, the respective Application Service provides status information to BP Enactment, thus allowing monitoring of the overall enactment.

Execution proceeds until either failures occur (like contract breaches, destruction of services etc. – cf. sections 5), 6)) or the business process is finalised, in which case dissolution of the VO is initialised (cf. section 7)).

4) Dynamic addition of an organisation during operation

Not all Application Service Providers are necessarily identified during the formation phase of the virtual organisation, as some tasks may only be performed after a comparatively long period of time and hence reservation of a service for that duration is unfeasible. Under such circumstances, the difficulty connected with the discovery process has to be considered, as some services are less common and/or are frequently occupied – assuming that the required service does exist at all.

Hence, such an approach is in particular sensible, if the required services are relatively common and only needed for short intervals.

The discovery process is either triggered directly by the need for a non-manned service arising or by a specific discovery-related activity in the collaboration definition. In the first case, the address of a non-existent service is requested from VO Management which in turn triggers the discovery process at BP Enactment, whilst in the second case the identification process reflects a separate task in the business process. Likewise, the latter case allows for discovering new services ahead of time, i.e. before they are actually needed, hence reducing potential delays in the overall execution.

Flexibility of discovery during the actual operation of the virtual organisation is limited as opposed to during the discovery and formation phase, since no changes in the parameters of other service providers can be accepted in order to achieve the overall goal.

Once an Application Service Provider has been identified, it is provided with the required configuration data, as described above (section 2)). All VO participants that need to interact with this new service are informed of the change, respectively of the addition of a new participant, by providing the contact details (including access authorisation), i.e. Endpoint Reference Address to them. This also applies to Trusted Third Parties services insofar as they interact with the Application Services (e.g. Message Brokering, cf. section III.2.b).

5) Dynamic removal of an organisation during operation

Similar to adding a service provider during the operational phase, an organisation may want to free their resources again, once they are no longer needed by the VO. Accordingly, the Application Service Provider has to be removed from the virtual organisation, if so requested.

Again, the request is either raised directly (in this case by the Application Service Provider him-/herself) or indirectly by the respective entry in the overall business process. In either case, the message is forwarded to VO Management, which triggers re-configuration as follows:

As the service provider has no further rights to access other services and should not do so for security reasons, all respective access rights are revoked. In order to avoid further communication and in particular forwarding of (possibly sensitive) notifications, all references to the respective service are removed – the only communication remaining takes place between VO Management Services and the Application Service Provider.

If a price for service usage was agreed upon, billing takes place at this time – the Log may serve as a means for establishing the actual price. Similarly, the trustworthiness of the service provider is updated on basis of its performance (as maintained in the Log), i.e. the reputation gained through participation in this virtual organisation is forwarded to the Trust Maintenance Service.

Finally, the service provider is removed from the list of VO participants at VO Management side.

6) Replacement of a participant by another during operation of the VO (Evolution)

During the operational phase of the VO, a particular service may need replacing, due to non-performance, contract breaching, simple “disappearance” of the service or similar reasons. In either of these cases, the overall business process is delayed as the current task cannot be executed. Since the need for substitution generally arises after the actual task started execution, replacement may even cause a rollback and compensation operation in the involved BPs, as a set of tasks will (in most cases) have to start anew with the new service.

Typically, a Policy Subsystem identifies the need for a replacement as a reaction to a specific event, like e.g. contract breaching and notifies VO Management. The latter may verify the correctness of data by directly requesting information from the respective Application Service Provider.

VO Management then triggers re-configuration of the VO as described in section 5) (insofar as the service is still available for contacting), i.e. it removes the service to be replaced from the virtual organisation.

At the same time, VO Management triggers the Discovery service to identify a new service provider that fulfils the criteria as defined for the one to be replaced. The Application Service Provider will then be provided with the necessary information as during the dynamic addition of an organisation (cf. section 4)).

Once set-up accordingly, i.e. the old service removed (all tokens and related information revoked) and a new service configured according to the VO’s needs, BP Enactment triggers execution of the Application Service. Since input data may have been lost during the replacement process, BP Enactment furthermore triggers the Application Service Providers representing the preceding tasks in the overall business to re-distribute their data to this new service.

Note that, similar to dynamic addition of organisations (cf. section 4)), circumstances like relevance of that service for the overall execution, availability of replacements etc. play a significant role in whether a service should be replaced. Data like amount of service providers initially identified for that role (during the discovery phase) may be crucial for further proceeding.

7) Dissolution of the Virtual Organisation

Once the overall business process has executed its final task[6], respectively destruction of the VO is triggered by VO Management (e.g. due to grave failure), the virtual organisation may be dissolved, i.e. all partners are removed from it as described in section 5).

Once all Application Service Providers have been detached, the configuration of the VO Management Services will be reset up to the point of reuse. This means that the VO Manager may decide to maintain e.g. the collaboration definition for later execution and keep a list of all service providers that performed well so that they may be contacted again.

Generally, however, we will consider the virtual organisation to be reset completely at this point, i.e. any new request to a VO Management service provider will have to start a complete new setup procedure as described in section 2).

I.4 The Participants' Model of TrustCoM

Virtual Organisations as envisaged by TrustCoM can be regarded as the coordinated collaboration between individual business entities that share a common goal – generally, we may claim that such a business goal is the business opportunity the swiftly formed virtual organisation seeks to exploit. Thus the expertise required from business entities, their limitations and the general requirements are implicitly defined. Entities participating in such a VO all contribute in a defined way to this goal and need to pool resources in order to perform their respective tasks, i.e. the overall collaboration may be highly interactive. The collaboration during the VO's operation phase involves the exchange of messages realising the aforementioned level of coordinated collaborative activities. Even though such VOs may provoke the impression to be static in realising the goal, the actual participants may constantly change their private configurations and even an entire entity may either be replaced or added and dispatched dynamically over time. While the former does not necessarily have an impact on configurations of the VO itself, the latter however does, requiring the ability that a VO is able to adapt to fundamental organisational changes. This allows for collaborations that are highly dynamic and in principle capable of adapting to changes in the midst of VO operation.

For TrustCoM, collaboration takes place between VO members which are, regarded just by themselves, outside the VO context, otherwise independent legal entities. They exchange messages to connect separate business tasks contributing to the VO goal which are encapsulated by individual web service implementations. From a high-level, global point of view, a TrustCoM VO may thus be regarded as a coordinated interaction between individual web services (providers). This global collaboration perspective is called the collaboration definition or, better known in the web service world, the choreography of the VO.

[6] Note that a virtual organisation may be maintained for more than one execution of the business process and that not all tasks are necessarily orchestrated by BPs..

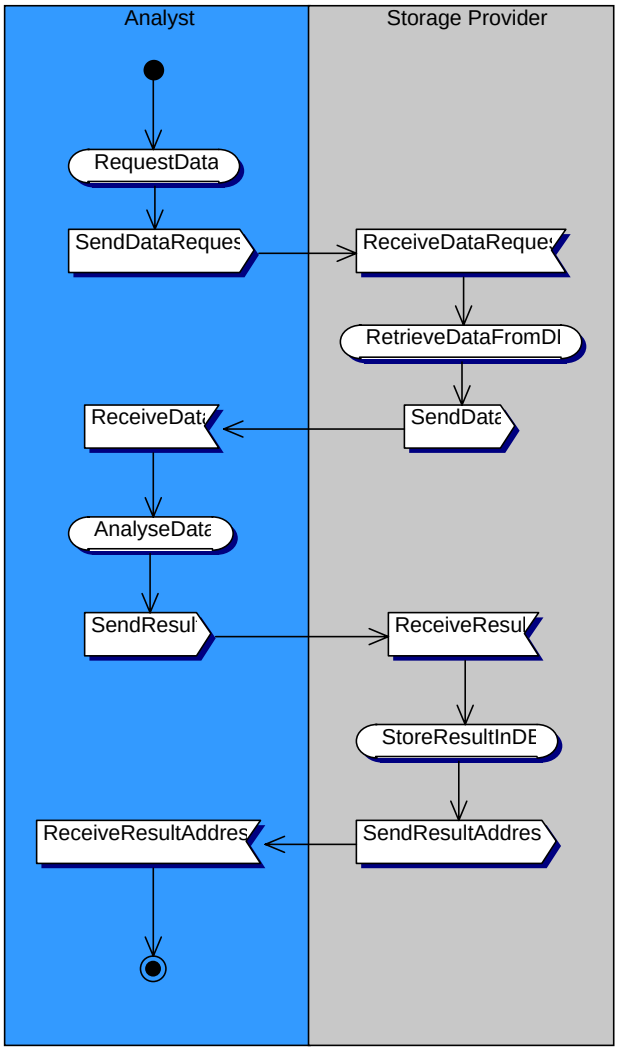


Figure 5: Sample collaboration definition as a UML activity diagram

I.4.a Business Processes

As shall be detailed in the following, we distinguish four “levels” of business processes according to their degree of abstraction, namely:

1. the goal description
2. the collaboration definition
3. the (individual) public business processes
4. the (individual) private business processes

The Goal Description

Every Virtual Organisation pursues a specific business goal as defined by a customer or VO initiator. Such a definition will generally be formulated abstractly without providing any details regarding how to realise this goal - in the case of the collaborative engineering (CE) testbed provided in WP35 to demonstrate the TrustCoM framework in operation, this goal may be formulated as "redesign and adaptation of an aeroplane regarding onboard

entertainment". The CE testbed scenario is set in the aerospace industry and details a plane maintenance and upgrade scenario. The scenario subset discussed here comprises only two participants, each playing one business role. The first business role is the one of a design analyst; the second is that of a storage provider. Notably this definition does not even detail the goal but nonetheless - as are described below - carries enough information to form a VO.

By making use of such abstract definitions, the virtual organisation will be allowed much more flexibility and dynamicity during execution whilst at the same time it releases any VO initiator from the requirement of having to know execution details: even though the initiator **may** specify a complete business process including all the details and requirements, we must generally assume that he or she lacks the respective expertise, thus also addressing the average customer as a potential initiator of a virtual organisation.

The Collaboration Definition

From the goal, an actual description of the high-level processes and the required business roles may be derived. This generally requires the help of some kind of “business expert” who knows how to define a collaboration definition and has a good understanding of what tasks are involved in the respective goals. The main contribution of the “business expert” is his knowledge on how to divide the work needed to be done to achieve the VO goal. This division leads to a separation of activities to business roles for which actors have to be discovered. For TrustCoM it is of no particular interest for the concept **how** the definition is derived from the goal statement – without loss of generality we may assume that such an expert either provides his/her support either as a web service or feeds a public repository with sets of potential collaboration descriptions for various goals (see Appendix, I.2b for details).

The actual collaboration definition covers the following main issues:

- a description of the involved business roles
- the requirements and restrictions
- high-level activities
- the interaction between these actors

This way, a collaboration definition provides not only all the relevant information for reaching the goal by specifying the sequence of interactions and data-exchanges, but also provides the relevant information for actually identifying the required actors, i.e. their description. TrustCoM extends this concept by adding some means of deriving the requirements from the overall restrictions as provided by the initiator – this covers e.g. how to calculate budget-limitations for each party given the available budget or individual time constraints on basis of the overall deadline etc.

For instance Figure 5 depicts a simple sequence of interactions between two business roles, a design analyst and a storage provider within a collaborative engineering scenario where they are respectively analysing aircraft designs and providing the storage to hold the analyses. The analyst is billed by the storage provider for storage space needed for a plane’s design data. The analyst performs analysis work on such data. To find an actor for the role of storage provider, the role of an analyst for instance imposes a budget restriction such as the storage space for the entire collaboration time period should not be more expensive than 3000€. A time restriction might be that the access time to the analysis data

should not take longer than 3 seconds. The latter would then result to a bandwidth requirement for the storage provider.

For TrustCoM, such a collaboration definition is a fully valid “collaborative business process” comprising the global view of the entire set of participants and their business roles and in general the only process related description the VO has to take care of on its highest level, even though it is **not** a fully detailed description of all involved tasks, interactions and as such not on the same level of detailed modelling as an executable business process. The latter is a means to enact a certain business role in a collaboration definition and it is up to the actor how to conform to the required sequence of message exchanges and related requirements and restrictions. This specific sequence will be referred to in the following sections as the business protocol.

The (Individual) Public Business Processes

Actors enacting the “business roles” in a collaboration definition conform to the required business protocol to reach the VO goal. While keeping their assets such as internal services and optimised processes private, they are obliged to at least expose the communication endpoints for participating in the business protocol. Figure 6 depicts this sequence of message exchanges with the communication arrows between the “AnalysisPartner” and “StoragePartner” swim lane. This diagram follows the line of the previously introduced example collaboration definition by illustrating the actual private and public business processes.

Public Business Processes are the conceptual components which facilitate the controlled exposure of only those endpoints. Those are not executable business processes, rather the interface layer to the latter – keeping private processes protected inside the own domain while allowing for collaboration in the VO.

The (Individual) Private Business Processes

Actors in the collaboration definition are not identical to tasks in a business process and accordingly the actors are not the actual web services, as shall be described in more detail in section I.4.b. An actor is a business entity, e.g. an organisation, company or department, which aggregates services. In fact, an actor may not even publish all the tasks, web services and resources that are involved in performing a specific “business role” due to privacy issues. Even though a business entity is free to do so, TrustCoM supports the issues involved in **only exposing those assets in a controlled manner which are necessary to participate in a VO**. Without loss of generality we hence assume that the actions defined in the collaboration description do not map directly to the tasks that are actually performed by the actor, i.e. the service provider plays a “business *role*” in the collaboration that implies the enactment of individual *tasks* **internal** to that actor.

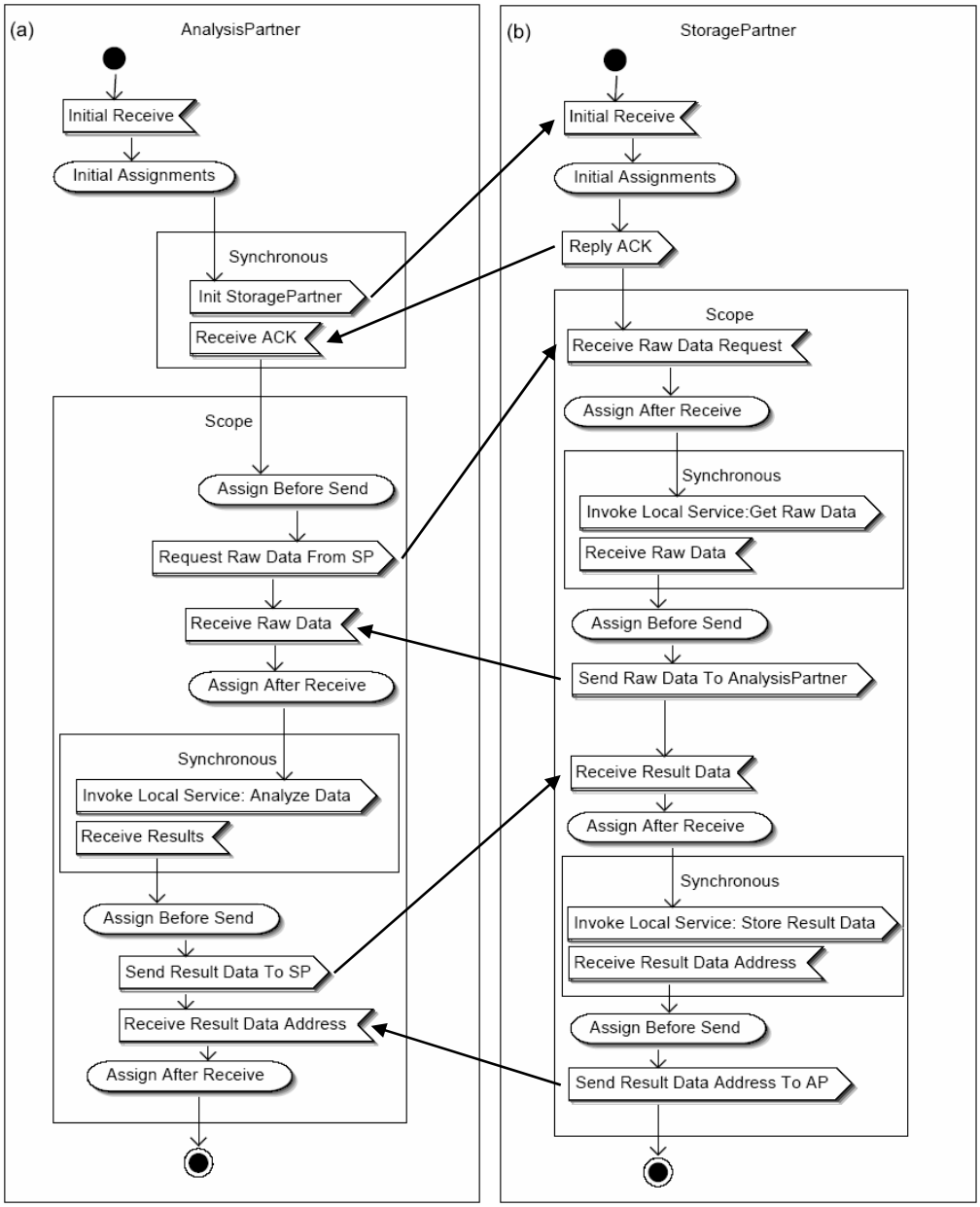


Figure 6: Private and Public Business Processes

To follow the above example, Figure 6 shows the private processes for each actor in a separate swim lane as a UML activity diagram. The “analysis partner” (business role), after agreement on an assignment, will actually have to analyse the existing design (set of activities in scope), e.g. analyse the requirements from the “entertainment designer” (business role, not shown here), based on the design data provided by the “StoragePartner” (business role). From the customer’s, as well as the other actors’ point of view, these details are of no importance however, and will unnecessarily complicate the overall collaboration description. Accordingly, this (individual) business process is internal to the business role and the execution details, e.g. a BPEL private process model, may be completely unknown outside the respective domain. Note however, that these business processes still have to comply with the overall requirements, e.g. if the customer explicitly

stated that no subcontracting will be accepted, the business process may not involve any tasks that have to be performed by actors that are not part of the VO etc.

In summary, the TrustCoM VO principally undergoes three phases regarding its "business description": (1) The customer or initiator provides a description of the goal that is to be achieved by the VO. (2) From this goal a collaboration definition is derived that specifies what kind of actors are required, which high-level activities they have to perform and how they interact etc. This information will support the identification of and negotiation with potential VO participants. (3) Each business entity that actually participates in the VO will "convert" the respective role descriptions and requirements into individual business processes that can be enacted it.

The business processes in (3) entail private and public processes. Since each business entity performs this step locally, the private knowledge of highly optimised business processes, e.g. efficiently retrieve and store design data, can be used for the private process. The public process needs to comply with the required interaction sequence and message types for participating with another business role, only exposing the required interface information while hiding the private process.

I.4.b Structure of Virtual Organisations

TrustCoM aims at realising Virtual Organisations on basis of web service interaction, thus allowing secure and coordinated interactions across enterprise boundaries. All actors in such a VO expose their functionalities through standard web service interfaces.

However, as has already been shown by such projects as GrASP, this does not imply that the actual processes can only be plain web services, but rather that all interaction between those are *exposed* as web services. This means that in principal anything that can or is linked to a computer can act as a participant in this VO model. This is of particular interest to TrustCoM as it has an impact on what we mean by "participant" and "(business) role" in a business collaboration:

Since collaborating partners are organisations who are more than just plain web services, we need to distinguish between the VO view on participants and their actual internal structure. The latter raises security implications regarding asset protection, privacy and data confidentiality as well as controlled exposure of the minimal required collaboration infrastructure. This relates to the distinction between collaboration definitions and individual business processes as detailed in the preceding section. Accordingly we need to clarify that even though we speak about (web) service providers interacting in a virtual organisation, it is really "business roles" that are realised / provided by the individual participants which again expose their functionalities as web services. From a VO-perspective there is no real difference however, whether the web service used for collaboration is just an interface to more complex executable processes or actually encompassing the business role's entire behavioural interface for the virtual organisation.

To allow full integration into the VO lifecycle, in particular to enable autonomous discovery **according to the collaboration definition**, we must assume that information about the roles the individual entities can fulfil have been published in the enterprise network. As the functionalities are provided via web service interfaces, this process will be principally identical to the one for publishing web services.

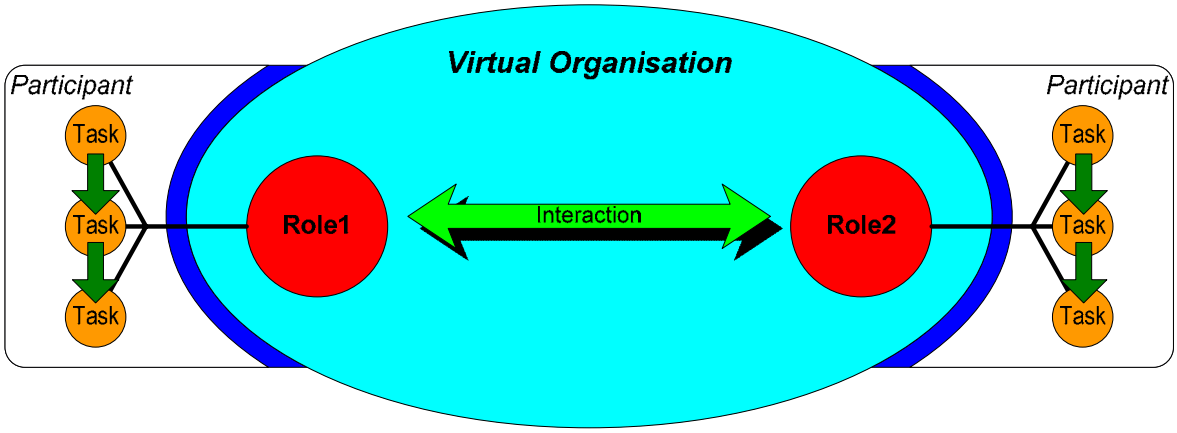


Figure 7: mapping between roles and actual tasks in the VO.

With this conceptual approach we ensure that any participant in the virtual organisation maintains full control over his/her resources and thus can enforce the respective privacy issues.

One needs to distinguish between services that are VO-aware and services that are not VO-aware. Not VO-aware services, an example could be an analysis service in the collaborative engineering scenario, can be offered to multiple VOs without any impact to the service conceptually. The business process engine and the TrustCoM framework will take care of the technical details that the service needs to adhere to. Any legacy service should be usable this way. VO-aware services, an example could be the storage service in the collaborative engineering scenario, are inherently adapted to the VO, e.g. by an identifier that corresponds to the storage place for the data. The business process mechanism can combine VO-aware and VO-unaware public and private services into a combined choreography.

Offering multiple services, products or similar to same VO does conceptually have no impact on the business process. Logically the participant can be seen as multiple participants, but since configuration and setup are automated, there is no real noticeable difference to the user.

A company can offer many variations in the form of public processes for the same private process. In that way the creator of the choreography can choose and control in which way he wants to use the private process and adapt the overall business process to the needs of the VO. An example would be to hardwire certain parameters to the web service call of the private process and offer those as different public processes. It is important to note that the participant can control this way how much control over the business process he will be executing he is transferring to the VO initiator. This implies that the control is tuneable to almost any degree the participant intends. In an extreme he could separate the steps in his private process and make all of them publicly available, thereby offering the VO initiator full control over the business process composition. On the other hand the participant could implement a full-blown large-scale business process as a private process and only offer its inputs and outputs once to the VO initiator.

I.5 The Trust Model in the TrustCoM Framework

Trust is usually presented as a positive and defining quality of good relationships, particularly where there are strong interdependencies between the actors. Thus trust is an essential aspect of relationships, for example between clients and contractors and purchasers and providers – the relationships established in the VO addressed by the TrustCoM framework. The immediate advantage of trust between actors is that it reduces the effort of maintaining all aspects of the relationship, allowing each to take some aspects simply for granted. However, the corollary of trust, or trusting, is the risk to which an agent is exposed by reduced vigilance. Trust management in the TrustCoM framework is the management of the tradeoff between the cost of vigilantly maintaining all aspects of the VO relationship and the risk of trusting actors in a VO.

Trust is an attitude of individual humans, and is applied only metaphorically to organisations or to objects; although often, more precisely, as a metonymy where the trust relationship holds between the individual senior managers of two organisations but is generalised to the whole organisations. Social scientists differentiate trust in a person from a judgement of competence in a person – the judgement that a person is competent to fulfil a role. The residual notion of trust is usually defined in terms of the commonality of intentions between two parties – that is, that another person is *on your side*. Trust becomes important on occasions when other supports to the relationship have broken down – for example when contracts are breached by error and the other party is willing to forego redress, or when unexpected circumstances occur in which a party is willing to make short term losses in order to maintain the relationship, in the uncertain expectation of receiving long term benefit. Across the population, individuals vary considerable in the variety, number and power of other supports to relationships. Consequently, they vary considerably in the ease and frequency with which they must rely upon trust alone to guide them. Within the computer mediated VO relationship, the TrustCoM framework is designed to provide both many supports that can maintain a relationship before relying on trust, and a basis for establishing trust itself before they fail.

A commonly attributed untrustworthiness in software is shown by web browsers that pop-up extra windows to advertise products that the user does not want, or which transmit personal data to another organisation against the user's wishes for privacy. In these cases the distrust of the software is generated by it appearing to act against the user's interests or intentions[7]. Obviously all notions of trust of a computer system are metaphoric, since it has no intentions itself, and the intentions of neither the creator nor the owner can be known, but only inferred from the behaviour of the system. Thus, for the software implementation of the TrustCoM framework the trust of the user in the software, and the organisation presented through the software will be both metaphoric and inferred from the behaviour of the system and the organisation operating through it. Since that has been stated, the term trust will be used from now on with reference to organisations and software without constantly noting its metaphorical nature.

Consequently, TrustCoM is developing a framework where the behaviour of the software and the organisation operating through it are explicitly constrained, are transparent, and

[7] Social scientists also address the complex case of judgements of trust when an individual acts against another's short term interests and intentions, while acting for their longer term interests. However, such cases are too complex for consideration within the present project.

provide a basis to predict future behaviour in order to foster trust in the user. In practice the mechanisms to constrain the behaviour are contracts and service level agreements (SLA) linked to collaborative business process models (BPM) which between them define what operations can be done, what access is permitted by whom and for what purpose, and what are the consequences of breaking the agreement – the operations and their context are clearly and securely defined. Transparency is provided by a publicly available agreement and BPM whose operation is implemented. Both transparency and the basis to predict future behaviour are provided by the monitoring of the performance of the BPM, recording the time and quality of performance according to the SLA, and drawing on this as a record to predict the competence of an organisation to fulfil its role[8]. The consequence of these mechanisms is that business risks are mitigated. Consequently, reliance can be placed on business partners because partners can be selected on the basis of a record of their past performance in a role, and each organisation will be informed as soon as they fail to be reliable, so that the risk can be managed.

Two further terminological points arise in this context. Firstly, the term used in computing research for recording historic performance information and using it to support decisions is *reputation management*. This term has unacceptable connotations in many fields, where such technologies are called “supplier qualification systems”; however the term reputation management will be used in this framework although the less worrying term can be substituted.

A second terminological confusion can arise from a specific use of the term *trust* in computing to refer to the method of transmitting trust. That is, a *trust technology* is one that transmits authority to trust the statements (tokens or certificates) of an issuer. Consequently, a trusted entity is one where authority to trust has been transmitted. This idiomatic restriction is perfectly consistent with the conceptualisation above although limited to avoid the complex issues of what trust is, or how it is brought about.

I.6 The Contract Model of TrustCoM

Virtual Organisations as envisaged by TrustCoM can be regarded as the coordinated collaboration between business entities that share a common goal. From a legal perspective, the virtual organisation will normally not be considered as an organisation with legal personality, but as an instance of collaboration between the VO members. The key means to steer this collaboration is a contract or a set of contracts between the participating organisations. These contracts play a vital role in governing commercial interactions between organisations. Moreover, the contracts need to be closely linked to business processes in the e-business applications. This interplay between the legal level and the business process level is necessary in order to facilitate the joint approach towards the achievement of the common goal and to reduce inherent risks. This integration is facilitated through the TrustCoM concept of General VO Agreement (GVOA). The GVOA is a “container” of VO contracts, SLAs and policies that all partners agree to.

[8] “A contractor's past performance record is arguably the key indicator for predicting future performance.” (US Department of Commerce and the Office of Federal Procurement Policy) but “a fund's past performance does not necessarily predict future results” (US Security and Exchange Commission).

A contract is often defined as a legally binding agreement that creates an obligation to do or not to do a particular thing.[9] In the context of the TrustCoM project, our focus is on the internal legally binding agreements between VO participants.[10]

Whether agreements between (prospective) VO participants can be considered as contracts, depends on whether the parties regard them as legally binding and enforceable. A contract is usually formed by an offer and an acceptance; sources of law (national or international) provide detailed requirements on contract formation.[11]

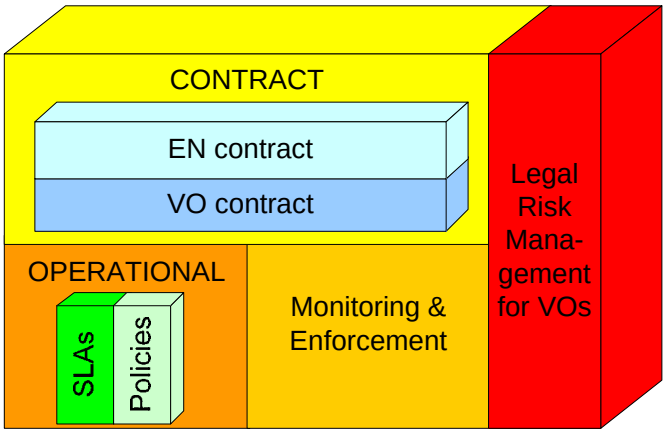


Figure 8: The Contract Model in TrustCoM

TrustCoM has followed an approach in which the GVOA includes:

- VO Contract: a contract that express the general rules that each partner of a VO must abide to. These general rules for of collaboration constitute the legal basis for the collaboration. They define how the VO collaborates towards the achievement of the common goal and how the partners jointly work with reducing the risks of collaboration.
- Business Process – a model of the process that the collaborative business will follow in order to achieve its common goal. Schedule to the GVOA.
- Service level agreement (SLA): specification of the performance that partners involved in a specific (operational) business process must abide to in order to perform their role defined in the business process. Schedule to the GVOA.
- Access Control Policies – permissions to access resources (computational, data, services) at times specified in the business process in order to perform the role defined for them in the business process model. Schedule to the GVOA as a mechanism to enforce confidentiality.

[9] William P. Statsky, *West's legal thesaurus/dictionary,* West Publ., St. Paul 1986. One may want to add that the contracts also may contain permissions.

[10] There will also be contracts involving VO members and third parties, see e.g. Report on Consumer Protection and contracting with 3rd parties by the ALIVE IST project http://www.vive-ig.net/projects/alive/Documents/Consumer_Protection.zip.

[11] As an example, consider the United Nations Convention on Contracts for the International Sale of Goods (CISG) Article 14 I (1) and Article 18 I, even though they address goods and not services.

A VO contract identifies and specifies the general rules that characterise how operational business processes are to be conducted through collaboration in a VO. On the other hand an SLA describes Quality of Service (QoS) objectives for a specific service as agreed by the service provider and the service consumer.

These contract types also need to be related to the different organisational levels of collaboration. The creation of VOs may be facilitated by an Enterprise Network (EN), which is set up as a basis for more specific collaboration in VOs. This EN will and should also be based on a contract which should include rules about the collaboration at EN level and about the creation of VOs. Hence, if there is a contract-based EN, both VO contract and SLAs may be understood within the context of the EN contract.

I.6.a EN Contract

The EN contract will be drafted by the EN founding members; it will be formulated in natural language.

A template for an EN contract is included in the Report Legal Issues in SME clusters, provided by the Legal-IST project (legal-ist.org). An EN contract should at least cover basic issues for collaboration, including

- EN structure
- EN governance (EN management structure, etc.)
- Outline of VOs (industry domain, VO management, etc.)
- IPR & confidentiality issues
- Data protection issues, if applicable
- Payment & Costs,
- Liability and insurance
- Jurisdiction & Choice of Law
- Dispute settlement
- Etc.

EN contracts will be defined by the EN, based on the types of VOs envisaged by the network, taking into account the specific needs of the industry in question and based on the requirements laid down the applicable national law. Though templates and model contracts are available, it is not possible to draft one general EN contract for all applications. There will be major differences between possible networks in various industries, services, jurisdictions, etc. The more similar the VOs in the network are, the more details may be included in the EN contract.

I.6.b VO Contracts

VO contracts may be written in natural language, but machine readable formats are used in parallel for those sections which the TrustCoM framework can support automatically. The content of VO contracts essentially depends on the specific kind of collaboration and on the relevant industry, (e.g. collaborative engineering in the aerospace industry or provisioning of eLearning services). A VO contract template in natural language was provided by ALIVE

IST project (consortium agreement type of contract). More specific model contracts for different contexts are available in legal literature.[12]

Amongst the issues to be addressed by the VO contract are QoS requirements, access rights to computational resources, and trust issues (including consequences of one VO partner's reputation level falling below a stated threshold).

A particular challenge in relation to VOs is the speed with which they may be expected to be formed, potentially on a time scale on the order of minutes. Creation and signing of VO contracts may thus need to be fully automatic.

The creation of VO contracts may be facilitated through the use of templates drafted e.g. at EN level. EN members from a certain industry (e.g. collaborative engineering) will normally have access to typical contract models utilized in their industry. Based on these typical contract models, VO contract templates can be drafted by the EN founding members. In cases where there are major differences between VO contracts in an EN, the EN may need to draft several different VO contract templates. Some of these templates may be very detailed, leaving only some specific matters (e.g. QoS requirements and price) for the actual contract negotiation. The degree to which the VO templates need to be adapted depends on how much the VO contracts differ from each other.

I.6.c Contracts and the VO lifecycle

The EN and VO contracts will also need to address the different phases of the VO lifecycle: Firstly, in the pre-contractual stage of the VO (identification and formation), there may be preliminary contracts (letter of intent, memorandum of understanding/preliminary contract) regulating the creation of the VO.[13] At the same time, the EN contract may include rules for the creation of VOs, e.g. regarding the selection of prospective partners, confidentiality duties, etc. Secondly, the operation as well as the evolution of the VO will follow the rules laid down in the EN and/or VO contract. Thirdly, the dissolution of the VO will need to follow the contractual rules, and VO contracts will typically include rules about the effects of termination of the VO contract.[14] A VO contract may e.g. include confidentiality duties which will prevail even after dissolution. Similarly, liability issues may need to be addressed after dissolution. Last but not least, if the VO is expected to generate results that may be IP protected, then the VO contract should address IP rights and use by VO members after dissolution. Hence, though the VO is dissolved, some contract provisions will remain valid and will require the attention of the VO partners. The contract should therefore be available for VO partners also after dissolution.

I.6.d Examples from the TrustCoM test bed scenarios

The TrustCoM test bed scenarios illustrate that there will be major differences between VO contracts in different industries: The eLearning scenario envisages that there will be a Metacampus EN contract, i.e. a rather stable contract for those participating in the

[12] See, e.g., Richard Morgan and Kit Burden, *Morgan and Burden on computer contracts,* 7th edition Sweet & Maxwell, London 2005.

[13] See, e.g. ALIVE IST Project *VE Model Contracts, Deliverable D 17a* (2002), Section 3.

[14] For further details see ibid, Section 4.6 on p. 27.

marketplace. In this scenario, VO formation needs to happen in a matter of seconds or minutes as the user requests and then selects a learning path via the portal. Since the VOs only differ with respect to the learning paths, the involved LCPs and end-users, most legal issues may be covered in the EN contract. Content providers could, for example, join the EN and agree to be bound by the EN contract when registering their services in the eLearning EN. Nevertheless there is a need for a (rather operational) eLearning VO contract that governs the provision of eLearning services to one end-user, based on one learning path. Contract templates could be specified e.g. by the initial eLearning EN founder(s) and agreed to by each EN partner as they join the EN. This would need to be anchored in the EN contract.

In the CE scenario, VOs will differ markedly from each other: Therefore, the EN will either be a rather loose club of collaborators, or there will be a multiplicity of ENs, or the EN is centralized around the CE VO. Nevertheless, there will probably be more stable contractual relations

- between the CEVO and the Air VO, on the one hand,
- between the CE VO and a group of (potential) service providers, on the other hand.

The VO contracts between CE VO and other participants will differ based on the type of contract, e.g. outsourcing, ASP, consultancy, software licenses, combined contracts, etc. Model contracts and guidelines for the different contract types are available in legal literature.[15]

I.6.e Drafting EN and VO contracts

EN and VO contracts will be drafted based on an assessment of the planned collaboration at EN and VO level. This assessment should both cover positive aspects (what is the business objective of the EN/VO and how can it be achieved) and negative aspects (risks related to the collaboration, affecting either the common business goal or the assets of the participants).

The drafting of some elements of the EN or VO contract will be based on the business plan and strategy, on the specific needs of the industry in question and on specific requirements laid down the applicable national law. This positive assessment will take into account the envisaged VOs the VO lifecycle, the VO management structure and what in TrustCoM is referred to as the collaboration definition. The collaboration definition includes a description of the involved actors, specified as business roles, and restrictions on such actors. This information constitutes the input to define a VO contract.

Moreover, the EN or VO contract needs to take into account risks related to the collaboration. This aspect can be covered in a Legal Risk Analysis, which seeks to identify risks related to the collaboration, affecting either the common business goal or the assets of the participants. These risks may be identified and analysed in a structured way. This analysis results in a list of risks, which may be prioritized according to their likelihood and consequence value. This risk assessment serves as a basis for the drafting of rules in the EN or VO contract. Moreover, legal risk management serves as a bridge between the

[15] Ibid.

contract level and the operational technological level, including monitoring and enforcement. In particular, legal risk management may be utilized in order to

- Identify risks that need to be taken into account when drafting the EN or VO contract, including risks related to policies specified as described in the TrustCoM framework;
- Identify issues of high importance within the operational part of the GVOA;
- Identify risk areas that should be monitored and rules that need to be particularly enforced.

I.7 Confidentiality and Privacy in TrustCoM

Both notions describe a situation of limited access to information, however the nature of the information as well as the values thus protected differ. Privacy is a right guaranteed to the individual with respect to personal information ensuring its integrity as a human being, autonomy or attentional self-determination. Commercial confidentiality on the other hand targets the information used in trade that affords the rightholder a competitive advantage. What it is protected is the effort invested in research or analysis. Confidential information may refer to trade secrets, know-how or to other information designated as party as confidential.

Confidentiality protection is more limited than other forms of IP protection, to the extent that a third party which develops independently the same technology cannot be prevented to use it, even against competitors who did get hold of the trade secrets in a derivative way from the original owner. Therefore, adequate access control policies are a must.

Due to the fact that trade secrets are not registered, the costs involved in the protection of trade secrets stem mainly from the requirement to put in place an information security and protection policy and program in the company as well as from monitoring, surveillance, audit and legal measures against those who breach or try to breach the security system. So long as a company has made systematic efforts that are considered reasonable under the circumstances to preserve confidentiality or secrecy, legal remedies are available in case of misappropriation of almost any kind of information of competitive value.

The disclosure of such information is optional, and those few business partners who do have access to it have to comply with restrictive conditions regarding their use and with an absolute prohibition of their disclosure to third parties. For example, the information exchanged by the parties during negotiations is to be kept confidential.

Larger operations such as those envisaged by the TrustCoM CE Scenario, involving numerous informational assets require in addition to a very broad collaboration agreement (a licensing agreement, a commissioning agreement, consultancy, joint-venture, partnership) a non-disclosure agreement expressing their agreement on each other's management and security standards involving the confidential information exchanged. The TrustCoM Deliverable D60 describes in detail what types of confidentiality clauses may appear into a confidentiality agreement and their role as treatments for the risks to confidentiality in the CE Scenario.

The parties cannot be forced to disclose confidential information at any point during the VO Lifecycle. However, there are few situations in which non-disclosure of certain information by the parties during negotiations may “destroy” the contract making it voidable and opening the possibility for the innocent party to claim damages. In such circumstances, the information cannot be hidden under the umbrella of confidentiality.

In the first case, the non-disclosure of relevant facts actively “tricks” the party into entering the contract. In the second case, one party is in error about certain facts and the other party does nothing to eliminate that error although he knew or ought to have known about it.

Both of these situations are seen as vitiating the consent given by one party in the conclusion of the contract. The party in error as well as the party that was “tricked” into entering the contract will be able to avoid the contract, meaning that he can request in court that the contract be seen as it has never existed, and even claim damages.

The contractual means of protection of confidential information offer limited or no protection at all in case a third party is at the origin of the disclosure. The confidentiality agreement between CE-VO and TC-ConsEng could not be invoked when TC-HPC is the one responsible for the confidentiality loss. Moreover, once the information reaches public domain the confidentiality agreement will have no value regarding it since either of the parties will be able to use the information as they please.

I.8 The Security Model in TrustCoM

In today’s work environments, employees have to fulfil many different roles and have to collaborate and work with a growing set of partner organizations. New collaboration models, such as the proliferation of the TrustCoM framework, accelerate this trend. For the people in the partner organizations, both managers and employees, the vast amount of partner organizations results in multiple challenges during the operational phase of virtual organizations:

First, employees no longer know all employees of their partner organizations personally, i.e., it is very difficult or even impossible to decide and manage correctly which partner organization employees need access to the company's resources. Imagine that collaboration partners would be forced to constantly communicate changes in their employee roles, such as "We have a new colleague, called Bob, who also needs access to this and that service". For TrustCoM, we've been looking into claims-based security models that allow a loosely-coupled and distributed security management inside the respective partner organizations, while having very little communications and coordination requirements. The main goal for us was: "How can we do identity, access and authorization management in a fully distributed way, while preserving the subject’s privacy to a maximum extend?"

The second challenge, both for the employee, as well as his manager, is the adequate protection of the employee's privacy during the fulfilment of his daily duties: How much (potentially sensitive and private) information about the employee needs to be communicated and shared with the partner organizations? On one hand, the partner organization needs assurance that the employee from the partner company is authorized to perform certain operations. On the other hand, the employee has a valid interest in a minimal disclosure of personal information, such as his claims or attributes, to the partner.

The security model we've developed in TrustCoM framework enables managers to define the claims that an employee possesses in a given virtual organization, while at the same time reducing the amount of additional personal identifiable information.

Another security aspect in the virtual organization's life cycle is the establishment of a virtual organization and maintenance of VO membership. We need to support different membership models, virtual organizations with rather static sets of partners, as well as virtual organizations where VO membership dynamically and frequently changes. Membership changes should be communicated on a need-to-know basis, i.e., partners that need to know whether a given partner is part of the virtual organization must be able to get fresh and reliable information about membership.

The complexity of the process of joining a virtual organization can vary significantly. The TrustCoM framework supports very simple join processes, such as a well-known VO candidate registering with a VO manager. Another join process could be an incremental and iterative negotiation and disclosure of properties between the VO and the potential VO partner.

II Conceptual Architecture

In order to realise a framework catering for the concepts as described in the previous chapter, it is necessary to break down the requirements into technical and non-technical aspects (like trust and legal issues). An architecture for middleware fulfilling the concepts would thus have to cater for the technical issues and at least support the non-technical aspects, where not providing a reference solution. As such, the framework will not be able to cover the *full* complexity of issues like trust and/or legal aspects, as many of these are not amenable to automated monitoring or enforcement at present – TrustCoM's interpretation basis is discussed in detail in chapter I.

The choices made for the TrustCoM architecture as described in the following section were strongly influenced by the business models (see section I.2), legal issues (see sections 1.6 & 1.7) and the requirements derived from the extreme use-cases defined by the two testbed scenarios (cf. chapter V). The approach resulting from these requirements is different from that that adopted by many academic research grid projects incorporating VOs which have not addressed many of these requirements - for example, they have emphasised technical issues such as performance on today's production service over the business and legal requirements that TrustCoM addresses..

Section II.2 will furthermore provide an overview over the models chosen for representing the architecture, so as to compensate for the typical shortcomings regarding flexibility of setup, respectively regarding comprehensibility.

II.1 From Concepts to Architecture

With respect to the conceptual reflections in chapter I, we can summarise that the TrustCoM framework has to respect the following main issues:

REQ_A
: In realistic business scenarios, a company will not provide individual resources, but rather the "products" they develop and sell as part of their business. [16] To produce these, companies typically execute their own (private) workflows that aggregate multiple local (and potentially outsourced) resources.

REQ_B
: The internal structure of a service provider is private, meaning that the TrustCoM framework will not modify or expose it. Instead, the framework exposes the relevant interfaces for obtaining (managing and influencing) the "products", in so far as the respective company allows.

REQ_C
: The service provider alone decides what information is available and how it is made available. Data considered confidential has to be respected.

REQ_D
: Each participant has his, respectively her own usage policies that have the utmost

[16] The language used in many countries does not make a distinction between *products* and *services* as the output of businesses. For those readers who do make this distinction, both are considered here.

priority and may by no means be overridden by the VO, even if this implies that the respective provider cannot participate in the collaboration.

REQ_E

Service providers may not want to support all functionalities necessary to maintain a virtual organisation by themselves, and/or may want to make use of already existing components rather than using the ones provided by TrustCoM. *Ideally*, a service provider will just choose the components on a “plug & play” basis, where it is completely up to him/her to “outsource” the functionality, choose own components or even skip it completely.

REQ_F

From the customer perspective, a Virtual Organisation should provide the “best” performance according to the customer specifications – this can range from actual performance values (QoS) over budget restrictions to the actual naming of providers.

REQ_G

Customers will in most cases not bring in the relevant business experience to steer and manage a Virtual Organisation, let alone to detail the individual business processes – the TrustCoM framework should provide an abstraction layer that allows interaction with the whole VO as a single *entity*.

REQ_H

Virtual Organisations may be created *on demand* and are *dynamic* with respect to changing environmental conditions / requirements.

REQ_I

In a VO, interactions partners are specified by the collaboration and not by the participants themselves – the framework needs to observe the fact that most companies will not take responsibility for other (unknown) parties’ performance (i.e. rely on them)

REQ_J

Service providers in a Virtual Organisation generally do not want to *depend* on other participants, in the sense of that the respective party may manipulate them

REQ_K

Similarly, in most business relationships defined by a third party (the VO), the partners will not *trust* the respective partner to *neutrally* evaluate and measure its own performance and/or pricing.

REQ_L

Sensitive data needs to be well protected and access to the local resources restricted to those instances that require this access.

REQ_M

The rules and policies of the Virtual Organisation and of each individual participant need to be enforced within the VO, so as to avoid failure, data misuse, security issues etc.

REQ_N

A record of each party’s performance should be maintained to justify management

actions taken within the terms of the GVOA if litigation takes place, and to provide a basis for the evaluation of parties' competence as participants in future VOs.

REQ_O
: Interaction takes place not only across individual organisational borders, but also across national borders, thus implying different legal situations, corporate policies and cultural backgrounds

REQ_P
: Service Providers may want to participate in more than one Virtual Organisation.

The following sections outlines the principles of the TrustCoM architecture and how they were developed to address these issues – this relates in particular to our choice of service types (section II.1.a) and the distinction between individual "subsystems" (section II.1.b).

II.1.a Abstract Structure

TrustCoM introduces a high-level VO structure that tries to accommodate for the main conceptual issues by distinguishing services according to the *type* of functionality they provide to the Virtual Organisation (cf. Figure 9). This implicitly serves as a classification with respect to the requirements and restrictions applicable to the respective Service Provider type (see also chapter IV). Note that this distinction has already been thoroughly discussed in previous documents (D09, ID1.1.3 etc.).

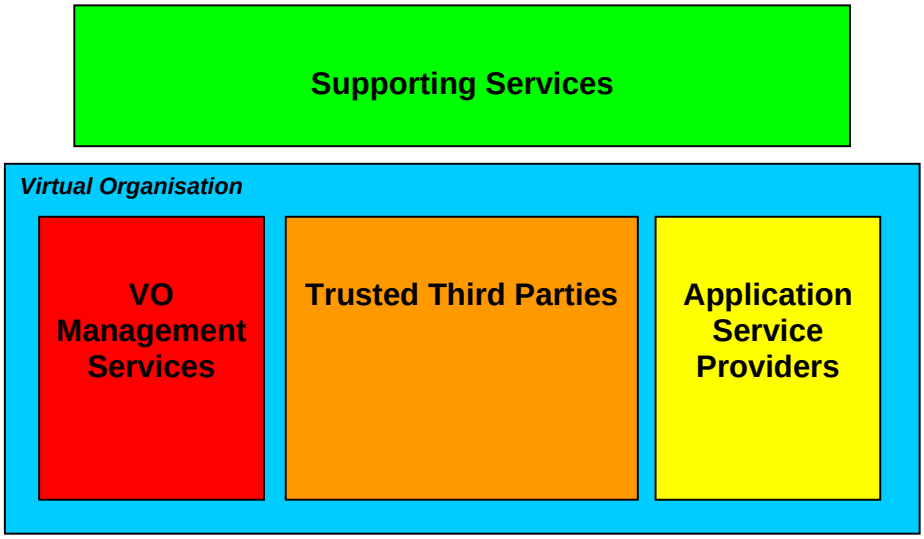


Figure 9: The service types participating in a Virtual Organisation.

- Application Service Providers (ASP)

Any entity that directly contributes to realising the VO's overall business objective as codified in the collaboration definition by fulfilling one or more roles in it, is acting as an Application Service Provider. These entities are Business Partners that are obliged to provide the respective services by contract and that demand payment for their contribution.

From the conceptual stance, ASPs are the main participants of a Virtual Organisation. For TrustCoM, these Service Providers expose only limited functionalities that are

directly coupled to the “product” they deliver rather than the resources they require for generating it (REQ_A, REQ_B, REQ_C, REQ_D)

- VO Management Services

These services provide the functionalities to coordinate the interactions between the VO services so as to reach a common goal, represented as a collaboration description. From a technical stance, no such single entity is required, as management related information may be equally distributed to all participants – however, a central coordination instance helps increasing flexibility (REQ_H) and manageability of involved parties (cf. REQ_I). This restricts the individual participants capabilities to manipulate the VO setup and hence each other (REQ_J, REQ_L, REQ_M).

The VO Manager (as a specific instance of these services) furthermore can act as the interface between customer(s) and participants of the VO – thus it represents the customer’s interests inside the VO (REQ_F, REQ_G, REQ_M).

- Trusted Third Parties (TTP)

Though Application Service Providers and VO Management Service Providers form the main actors in a virtual organisation and in fact would suffice for enacting collaborative workflows, additional types of services are recommended to supplement the TrustCoM specific functionalities. This relates in particular to functionalities that can not or should not be realised by the Application Service Providers – either due to the ASP wanting to “outsource” the functionality (REQ_E), in order to maintain privacy issues (REQ_L) or to reduce dependencies between ASPs (REQ_I, REQ_K):

  - function outsourcing

    though generally not recommendable, an ASP may leave certain management related functionalities, including the enactment of local policies, monitoring performance and similar issues (cf. below), up to third parties that he/she trusts to perform the respective tasks. This may also involve functionalities that the ASP would like to make use of, but can not realise him-/herself, like logging etc.

  - privacy enactment

    some participants may furthermore want to remain incognito for respective collaboration parties and as such may request means of brokering interactions through a third party trusted by him/her

  - “neutral” parties

    finally, ASP and/or VO Initiator may not trust the respective other to perform certain tasks neutrally, i.e. without cheating if profit for oneself may be gained from this – this applies in particular to supervision of performance and its relationship to payment. Shifting such responsibilities to third parties trusted by both ASP and VO Initiator equally will ensure that e.g. maintenance of the performance log is performed neutrally, without preferring the one result over the other.

Likewise, TTP services maintain data that may be confidential, making them subject to VO policies, agreements and in particular security issues. Accordingly, we consider TTP services as participants of the Virtual Organisation, since their behaviour is influenced by the requirements of the individual VOs.

Notably, all TTP functionalities *may* be enacted completely by either VO Management or Application Services, thus rendering the structure of TTPs very individual to each VO and not a general requirement, but a recommendation. In summary, Trusted Third Parties take an intermediary position between VO Management as a representative of the customer and the individual Application Service Providers as the main VO participants.

- Supporting Services

As opposed to Trusted Third Parties, there will be services involved in realising the VO's capabilities that in themselves do not directly participate in the VO, i.e. that provide the same unaltered functionalities to each customer, respectively VO. As opposed to this, Supporting Services do not directly participate in the VO, yet indirectly contributes to its functioning. In general, this concerns repository-like facilities that maintain lists of services (as potential VO participants). These types of services principally already exist (e.g. UDDI) and are not, respectively only minimally influenced by a virtual organisation.

According to this structure, Application Service Providers do not have to rely on other ASPs, but rather the management and sensitive functionalities are shifted to instances principally neutral to the individual participants. There is no restriction to all these functionalities actually being provided by ASP participants, i.e. any TTP or VO Management Service itself may be an ASP and the other way round.

One of the direct implications from this categorisation is that different degrees of adaptation requirements exist for the individual types of services: whilst generally an application service provider would want to make use of the full trust, contract and security features, trusted third parties will not necessary require full support, but mostly security control. Supporting services on the other hand are generally not impacted by these functionalities and will not integrate any of these features. This also applies to the communication layer – see section III.2 for more information.

II.1.b The Subsystem Segmentation

From the requirements and concepts as detailed in the previous sections, we can derive a set of functionalities that need to be provided by the TrustCoM framework that may be categorized as described in this paragraph and that reflect the individual expertise of the consortium. This categorisation is pursued throughout the project, so that individual functionalities, provided as services or components, are realised as part of the respective category, in the following also called "subsystem" of the TrustCoM framework.

With the Service Oriented Architecture approach pursued by the project, the individual components developed within the scope of the respective subsystems are principally usable in a stand-alone manner, as detailed in the following chapters. Since each subsystem reflects specific types of functionalities and hence requirements, a service provider may subsequently select individual components from each category, according to his/her needs. This approach was pursued to the degree that allows fulfilment of the business models as discussed in section I.2.

Note that each of these subsystems is described in more detail in the appendix to this document. The requirements given in brackets specify the conceptual issues addressed (directly or indirectly) by the according subsystem (cf. also overview in section II.1.c)

VO Management (REQ_F, REQ_G, REQ_H, REQ_I, REQ_J, REQ_L, REQ_M, REQ_O)

The VO Management component defines and stores details of each virtual organisation participating in the Virtual Organisation. It is divided into three main modules responsible for (a) lifecycle changes to the VO ("VO Lifecycle Management"), (b) maintaining the participants in the VO ("VO membership management") and (c) managing the General VO Agreement ("GVOA management"). These modules interact mainly with the SLA management component which creates and manages the detailed SLAs, and with the Business Process Enactment and Orchestration component which defines the business process of the VO, and enacts when the VO is in operation. The VO management component builds upon the data in the Enterprise Network Infrastructure when a VO manager wishes to create a VO in order to allow the business process of the VO to be defined in the BP manager along with the roles of potential organisations, it calls the EN Discovery Tool to discover candidate partners from within the Enterprise Network, calls the SLA Negotiator to negotiate with a candidate partner the details of the SLA to perform a role in the VO, then composes the legal General VO Agreement to be signed by all partners. Once the VO is created, the VO manager calls the BP Manager to enact the business processes of the VO. While the VO is in operation, the VO manager responds to evolutionary changes in the VO, as partners succeed or fail to meet deadlines, quality and other policies from the SLA, ultimately identifying replacement partners and renewing the GVOA. When the VO terminates the VO manager closes the VO down.

Business Process Enactment and Orchestration (REQ_A, REQ_B, REQ_C, REQ_G)

The subsystem catering for Business Process (BP) Enactment and Orchestration provides generic, flexible services to be used in different application scenarios as well as for VO Management related purposes. This subsystem provides autonomous functionalities implementing the three phased Collaborative Business Process modelling methodology which was defined in WP2/21.

BP enactment begins with the global view/choreography of the VO business objective, the process and the roles required to achieve a set of goals, encoded in the collaboration definition. The CDL++2BPEL service component takes the collaboration definition as input and following a top-down approach, derives process views and optionally private processes from it. The latter occurs if no pre-existing private processes have to be taken into account.

A BP Management service offers runtime management methods for the BP engine. This service allows for automatic deployment of derived BPs and views, as well as their execution. Associated with engine comes a monitoring component.

On top of the operational aspects of BP creation and execution, this subsystem also takes care of Trust, Security and Contract (TSC) Management controls for BPs. Such aspects may be assigned at design and runtime as TSC extension roles to design time artefacts called TSC tasks in BPs (see Appendix, section I.2 for details).

SLA Management Services (REQ_B, REQ_C, REQ_F, REQ_K, REQ_O)

The SLA Management subsystem provides a set of services that allow autonomous observation of individual service providers' performance and comparing these to a set of previously agreed upon quality of service parameters.

Accordingly, the subsystem needs to provide the functionalities to

a) negotiate SLA terms that meet both the service consumer's expectation with respect to the quality of service, as well as the service provider's capability (and intention) to maintain these.

b) monitor the performance of a specific service and/or its respective environment (like the host system's status)

c) compare the monitored information with the terms agreed upon during negotiation of the SLA.

A member of the Enterprise Network uses this subsystem to associate SLA templates with the services it may provide to an eventual VO. A potential consumer of the application service uses this subsystem to negotiate and sign an SLA with the service provider. The SLA Management subsystem assists VO Management (via the Discovery Service) in the search for services that can meet the QoS requirements of the VO.

Upon SLA violations, the SLA subsystem generates notifications that can be picked up by the Policy Subsystem in order to apply the proper adaptation policies.

Trust & Security Services (REQ_C, REQ_J, REQ_L, REQ_N)

The subsystem for trust & security services provides services related to the establishment and maintenance of trust relationships with a priori unknown partners from foreign security domains.

Establishment of trust relationships is provided by Security Token Services that can issue and validate security tokens across administrative domains, and corresponding configuration management services that can be used to adapt the local security configuration to dynamic changes in the VO.

Maintenance of trust relationships is provided by a reputation management services that collects individual ratings about the prior behaviour (reputation) of Enterprise Network members, offers a combined reputation value to interested clients, and notifies registered VO members about sudden changes in this value due to recent activities. Also, a Secure Audit service provides the functionality to record custom data, for example actions performed by other partners, so that it cannot be repudiated.

Policy Control (REQ_D, REQ_F, REQ_H, REQ_L, REQ_M)

The policy subsystem provides the means to define, deploy and enforce both access control and adaptation policies within the TrustCoM framework. Access control policies comprise both authorisation policies that define which entities are permitted to access services within the TrustCoM framework and under which constraints, and delegation policies which specify permissions on the delegation of administrative permissions. Adaptation policies (traditionally sometimes called obligation policies) are in the form of event-condition-action rules that define how the VO should adapt in response to failures, changes in the reputation or performance of its participants, security threats etc. For example, policies would typically dictate under which conditions the procedures for the removal of a member of the VO should be executed in case of repeated VO breaches or significant loss of reputation. Similarly, policies can be used to trigger reconfiguration of the service message interceptors in order to add additional handling procedures such as secure auditing. Policy control is based around two services: the policy service which receives policies from the GVOA, deploys access control policies to the Policy Decision Point (PDP), enforces adaptation policies and manages the policy life-cycle and the policy decision point which enforces the

authorisation and delegation policies and responds to access control queries issued by the Policy Enforcement point which is part of the EN/VO infrastructure (cf. Appendix).

EN/VO Infrastructure (REQ_A, REQ_B, REQ_E, REQ_H, REQ_J, REQ_P)

Each service provider has his/her own approach to making the functionalities of the offered service(s) available, to managing them and to support trust, security and contract managing features – if any. The EN/VO Infrastructure components provide the base functionalities to allow common access and management functionalities across all participants in a virtual organisation. This involves in particular:

a) establishing a communication layer that allows messaging and notification, and relates the additional TrustCoM functionalities (trust, security and contract) to the respective services

b) maintaining the actual locations of services and mapping handlers to them so that services are accessible even when moved

c) supporting coordinated instantiation of involved services

d) exposing functionalities of services for discovery and

e) supporting discovery over a range of service-related information (WSDL, SLA etc.)

Thus the EN/VO Infrastructure may be regarded as providing TrustCoM's base layer.

II.1.c Conceptual Architecture Summary

This chapter summarises how the abstract structure introduced by the TrustCoM framework addresses the conceptual issues identified in section 1.

requirement	addressed how
SPs provide abstract "products" (REQ_A)	Each ASP is treated as an entity encapsulated behind a "gateway" like interface that allows exposing "virtual" function calls that bind e.g. to a local workflow. This enables Service Providers to react to (VO specific) requests in their own way, e.g. by triggering the relevant processes for manufacturing the according products. From the VO perspective, SPs are treated as "abstract entities" [20] that expose functionalities non-regarding their actual resources – accordingly, the overall workflow treats these as "products" rather than resources.
The SPs infrastructure is private (REQ_B)	The "gateway" like approach hides all SP specific (private) information up to the degree where necessary to enable interactions – however, the structure allows that virtual endpoints need to be exposed that provide the required functionalities without directly mapping to actual resources in the infrastructure. The security subsystem enhances these capabilities by restricting access to authenticated services with permission granted from VO and SP. The SLA subsystem too caters for confidentiality issues with respect to QoS related information by allowing the introduction of another level of abstraction through the gateway interface (cf. e.g. [19], [21]).
SPs define level of confidentiality (REQ_C)	Similar to the issues above, the "gateway" may be configured freely by the SP hosting it – this means that the exposed methods, the contract terms and the access rights / security settings are up to the administrator's discretion.
"Local" policies supersede VO requirements (REQ_D)	SP specific policies take influence in two ways: once during negotiation of the VO policy details and two, more specifically, the gateway structure allows for additional definition of local policies that are enacted prior to the VO policies. Given the

	specifics of the gateway, any local "policies" (like e.g. security configurations) will not be superseded by its functionalities, i.e. the gateway can not nullify existing policy means.
"Plug & play" usage (REQ_E)	The main point behind the gateway structure consists in allowing easy extension of given infrastructures with TrustCoM specific capabilities, thus reducing the impact on local resources – for example the message redirection support allows local services to interact "virtually" with any other resource without having to bother about its actual location. Furthermore, the Service Oriented Architecture approach pursued by TrustCoM allows easy extensibility and flexibility regarding the components, so that functionalities may be activated, respectively deactivated simply by (un)deploying them. In the same way, the TTP concept allows for easy outsourcing of functionalities (in so far as sensible, cf. chapter IV).
VO meets customer requirements (REQ_F)	The VO Management service acts on behalf of the customer inside the Virtual Organisation – its task consists in identifying the relevant providers to meet the customer requirements from both functional as well as non-functional (such as reputation, jurisdiction) perspective. The Policy Services ensure that the VO as a whole follows the according rules and conditions as defined in the choreography. The SLA Management subsystem supervises behaviour in particular regarding the Quality of Service locally to each SP and reports, respectively enforces performance specific issues.
The VO acts as an entity (REQ_G)	The main tasks of the VO Management service consists in setting up and managing business entities in a way that they form a dynamic, organised Virtual Organisation – in other words, the VOM plays an essential role in steering the participants in such a way that they form a (collaborative) entity. The whole collaboration is encapsulated by the VOM service in such a way that allows a customer / manager easy interaction with it on an abstract layer, i.e. without having to know the details about the data exchanges involved. Since participants are regarded as abstract entities, rather than sets of resources that need to be coordinated, the collaborative description is much more comprehensive and abstract than when full interaction details (and hence business expertise) would be required.
VOs are dynamic and created on demand (REQ_H)	With the more centralised approach of VO Management, membership information is more easily maintained, updated and applied, thus allowing for faster dynamic management of the VO structure. In combination with the Policy Service support, the VO can be quickly adapted to changing conditions both within the VO, as well as in the environment (such as business conditions). The ENVO support of TrustCoM allows easy message redirection and structure updates without affecting the infrastructure and thus the Service Provider.
SPs do not want to have to rely on each other (REQ_I)	In TrustCoM, contracts (and SLAs) are not formed between individual peers thus pushing responsibility for participants' behaviour upon the interacting parties. Rather, SLAs and contracts are formed between the (central) VO Management and each participating party, thus giving VO Management not only full responsibility, but also full legal power over the collaboration.
SPs may not manipulate each other (REQ_J)	Just as with data protection per VO participant, the Trust & Security components restrict unauthorised access attempts according to the SP administrators' specifications. In particular with the encapsulation provided by the EN/VO support, the actual resources are hidden from the other participants. Only VO Management

	related components may influence the gateway according to the VO requirements (such as altering redirection information), even though only insofar as the local policies allow this (cf. REQ_D). Further configuration specific data, like the (local) SLA is additionally protected through additional means of indirection (not directly exposed), so that – once activated – they may not be altered (unless complete reconfiguration, e.g. during re-negotiation) is triggered.
Evaluation and similar support is performed neutrally (REQ_K)	The TTP concept foresees easy outsourcing of components according to the Service Oriented Architecture approach. Accordingly, functionality considered "critical" from the stance of neutrality may (and should) be shifted to neutral Trusted Third Parties. As an example, the SLA Management subsystem considers the evaluation capability an independent component, respectively feature that is preferably hosted by a TTP.
Data and resource protection (REQ_L)	Data and resources are protected in multiple ways (cf. REQ_J): All message transaction is encrypted using the participants own security token only readable by the respective interaction partners (updated dynamically) Access is furthermore restricted and refined using local policies at the gateway structure. These policies are defined through the collaboration description and by the SP administrators' own discretion (cf. REQ_D) Resources and the general infrastructure are hidden through (and thus protected by) virtualisation
VO policies are enforced (REQ_M)	VO specific policies are derived from the collaboration description, the user requirements and potentially adapted through negotiations with the individual Service Providers. These policies define the conditions and terms applicable to VO behaviour, such as under what circumstances security needs to be increased, what measurements to take when a participant's reliability drops too low etc. The VO Policy service is part of VO Management up to the degree that it can take immediate measurements (like informing individual participants about non-performance etc.) itself, though generally enforcement will take place via the VO Management service by according triggers.
Reliability is measured (REQ_N)	The Trust related support of TrustCoM links to the SLA Management subsystem and its according performance information, so that contract related behaviour may directly influence the respective party's trustworthiness. As such, non-fulfilment of the negotiated Quality of Service may e.g. reduce the party's reliability. Such trustworthiness related information is maintained as "VO independent" and may thus feed back to multiple collaborations and provoke according actions. TrustCoM itself makes use of this capability in relationship to the VO specific policies (see REQ_M).
International cooperation (REQ_O)	Collaboration across national borders does not in itself pose a technical problem thanks to internet based messaging and standardised communication interfaces. However, international collaboration poses particular issues on the applicable jurisdiction in cases of contract breaches etc. – due to their nature, these issues can only be addressed marginally by the architecture, in so far as they do not require human interaction, namely of a lawyer. The main subsystems affected by this issue consist in VO and SLA Management. To address this circumstance as best as possible, we distinguish between SLAs as concrete technical descriptions of QoS to be maintained, Policies as means of describing the potential consequences from performance related behaviour and

	actual Contracts (GVOA) that define the legal basis.
Multiple VO participation (REQ_P)	The gateway allows for VO specific redirection of messages, as such the same gateway structure (and hence deployment setup) may be reused for participation in multiple VOs. The TrustCoM framework provides the relevant support for instantiating and configuring the according components and linking them to the gateway, respectively to the VO.

II.2 The Architectural Models

It has already been discussed that the framework cannot easily be represented by a single diagram, as it (a) involves too many components and in particular (b) since this will not depict the flexibility that TrustCoM provides.

Following a strong Service Oriented Architecture approach, the framework allows for a flexible setup not only of the Virtual Organisation, but also of each individual participant regarding the components (functionalities) to be supported, respectively already in place. As such, any diagram depicting actual distribution of components within the Virtual Organisation would implicitly represent just *one potential* VO structure, neglecting all other possible setups.

To overcome this issue, we introduce two types of description, each highlighting individual aspects of the overall framework, so that the full model may easily comprehended when examining these two descriptions, without requiring the help of excessively large diagrams:

- Relationship Model

  The relationship model depicts the dependencies between components, respectively systems in the framework. As opposed to a full deployment diagram, the relationship model does not convey any information about the *distribution* of components within the Virtual Organisation, nor does it give an insight into their *number*, i.e. how many of the respective type are instantiated - this way, the model is reduced to a minimum size.

  The main task of this model is to provide an insight into what *type* of components are functionally required for realising specific functionalities within a Virtual Organisation and implicitly which interfaces it has to realise itself in order to be integrated.

- Deployment Model

  As opposed to the relationship model, the deployment view provides insight into what components are *typically* deployed at which location for specific service provider types. Such a view will allow individual service providers to choose components according to their profile and respective needs. As opposed to the relationship model, it does not give any information about the interactions and only minimal information about the dependencies between components.

Both models together allow any participant in a Virtual Organisation to decide, which components are functionally required, where they should be deployed and what functionalities a substitute should principally provide. In combination with the Profile information of chapter VI, the necessary interfaces can be defined that each of the components should expose. Even though architecture and implementation are in-line, the profile information will go beyond the capabilities of the reference implementation (in the sense of providing recommendations rather than actual schemata, see there) - hence, care

needs to be taken when replacing available TrustCoM components with own services, so as to ensure compatibility.

III The Relationship View on the Architecture

Within this chapter we will present the relationship view on the architecture (cf. section II.2). This kind of diagram does not convey any information about the actual distribution of services and/or components within a Virtual Organisation and as such does not reflect the VO structure regarding which components belong to VO Management, Application Service Providers, Trusted Third Parties or Supporting Services. Implicitly, the *number* of instances is not reflected either, meaning that components like the Notification Proxy, which are principally deployed at every participant in the Virtual Organisation, are represented only *once* in the diagrams with a relationship to themselves where it comes to actual exchange of notification messages - in other words this means that Notification Proxies will take (amongst others) notifications as input and as output. This applies to any component in the diagrams meaning that every component will be represented only once.

As such, this view on the architecture does provide information about the functional dependencies between components. Accordingly, this model gives insight into

1. which components are required in order to realise specific functionalities
2. which components relate to each other and as such should be deployed “together”
3. the definition of the interfaces for potential substitute components

Note that the full description of each subsystem’s components may be found in the Appendix to this document. Here we do not detail the individual protocols but first of all want to depict a fully integrated view on the middleware.

Figure 10 below depicts the relationships between the *subsystems* (cf. section II.1.b) in a Virtual Organisation and as such provides a very high-level view on the TrustCoM middleware. It will be noted that the coupling between subsystems is comparatively loose as opposed to the one between components (cf. sections III.1, III.2) - this is simply due to the fact that subsystems realise individual logical functionality types (like policy management) and as such the data exchange in-between is limited to status information almost only.

It can furthermore be seen that there is a very VO Management centric dependency of all subsystem which reflects TrustCoM’s approach that in order to realise secure and reliable virtual organizations, some central management is required. Note that since the relationship diagrams do *not* convey deployment information this does *not* imply whether VO Management is realised as a single central instance or distributed across participants. However, as was discussed in chapter I and will be depicted in chapter IV, we do generally recommend a single, central instance for reasons of reliability, manageability and security.

The diagram does only represent the relationship between *subsystems* and as such does not convey information about the data dependencies between components per subsystem, as already mentioned above. Since the full view on all relationships within the TrustCoM middleware would not only exceed the available space but would also be too complex. To allow for a better overview and easier understanding, we hence split the full view into separate diagrams aligned to the main lifecycle phases, respectively main VO scenarios, as depicted in section I.3. Accordingly, the relationships as represented in the following diagrams are restricted to lifecycle phase specific issues.

Since most of the components supporting the EN/VO Infrastructure functionally contribute to almost all subsystems and, what is more, participate in many interactions as an intermediary, e.g. for message redirection, these specific relationships have been moved to a separate section (section III.2) in order to avoid confusion.

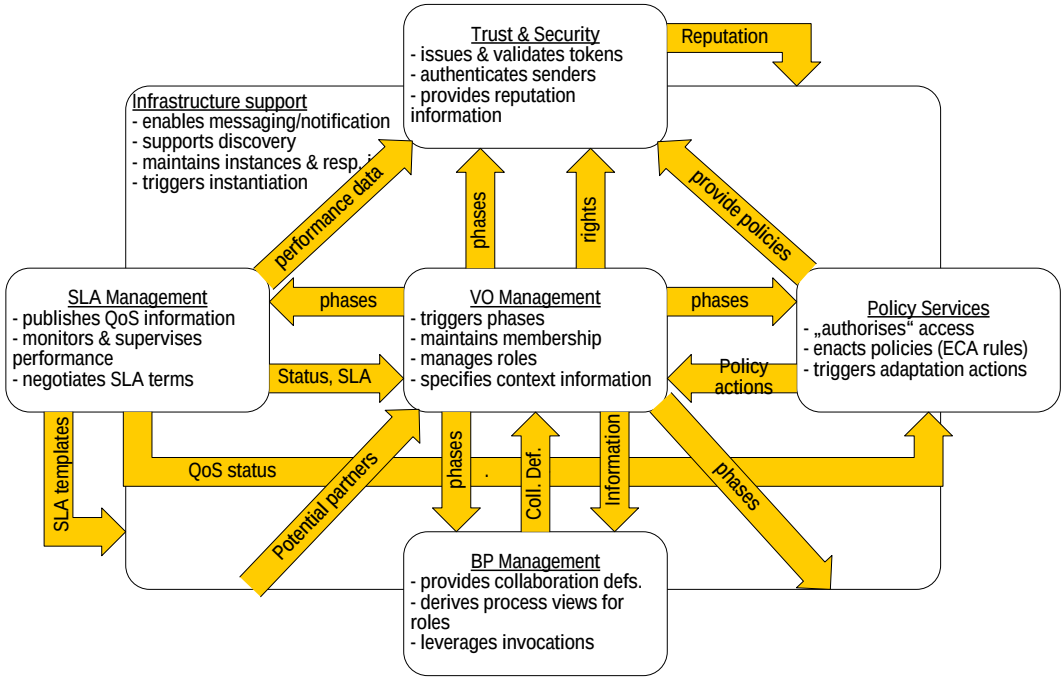


Figure 10: Relationships between subsystems in the TrustCoM middleware

III.1 The Relationships with respect to the individual VO lifecycle phases

Each lifecycle phase of a Virtual Organisation is centred on a specific task that the participants and the management of the VO contribute to in order to fulfil it - accordingly, the data exchange between components relates to the respective task of the phase only - the relationship diagrams below have been aligned to these functional distinctions.

The relationships between two components represent the information they exchange in order to realise their respective part in the lifecycle phase's tasks. Notably, the diagram does not depict the "full" message exchange, i.e. the request-response-pattern, but solely which component requires what information and how it may be provided. As such, the *sequence* of interactions is not encoded in the diagrams below, yet may be easily derived from the requirements per task and is described in a bit more detail in the accompanying text in each subsection.

Appendix B: Subsystem Architecture to this document provides a more detailed overview over the components and their functional relationships *per subsystem*. As opposed to this section, the appendix also describes sequence and messaging details between components for specific interactions. More details regarding the actual message structures can be found in Appendix A: Profiles (cf. section VI).

With the components related to supporting the EN/VO framework mostly missing in the diagrams provided here (cf. above), the relationships do not give any information about

whether the actual message exchange is of a direct request-response type or of an indirect, notification-like type. However, since any direct message exchange may be easily realised using notifications, this detail is not of direct importance for the middleware. The respective choices made by the consortium base on economical considerations rather than on technical requirements - these considerations take issues like message distribution and event-based message creation into account. Please refer to the TrustCoM Framework Appendix for messaging details with respect to the individual components.

The diagrams below depict components (respectively services) as circles that interconnect by arrows representing the information relationship between them. Plain dataflow is represented as arrows with black filled heads (→), whilst indirect relationships (e.g. through other non-depicted components) is depicted by arrows with white heads (⇢). Furthermore, arrows with open heads (→) denote "action" relationships, i.e. in particular triggers – this differs from the information relationship insofar, as the invocation message is generally static (e.g. with an empty parameters field).

Boxes with rounded edges that contain individual components stand for the subsystems subsuming the respective functionalities (cf. section II.1.b). Orange arrows (➔) signify the component relationships on a more abstract subsystem level. Figure 11 shows the "colour" coding used for the following diagrams by which components can be distinguished according to the subsystem they belong to. Note that most figures convey this information also by use of the rounded boxes.

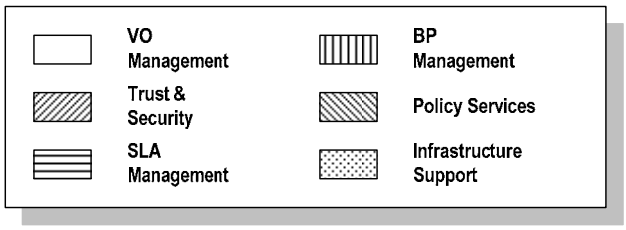


Figure 11: Legend for the diagrams in this chapter

III.1.a Preparation

Preparation for participation in a virtual organization is not really a lifecycle phase in itself; rather it reflects the necessary steps to take in order to participate in virtual organizations as envisaged by TrustCoM. Such steps involve mostly registration and publication processes in order to make the provided services / resources known and hence accessible. Thereby it is of no direct implication for the TrustCoM framework whether these repositories are within Enterprise Networks, i.e. where additional requirements have to be met by the services in order to get registered, or whether these are publicly accessible, like the UDDI repository by IBM[17] and SAP[18].

With Figure 12 we depict only the most recommended publication processes disregarding potential additional steps as required by the individual Enterprise Networks - as these will have no direct impact on the TrustCoM middleware as opposed to the ones presented here, this does not limit the applicability of our approach.

[17] Currently discontinued, see http://www-306.ibm.com/software/solutions/webservices/uddi/

[18] http://udditest.sap.com/webdynpro/dispatcher/sap.com/tc~uddi~webui~wdp/UDDIWebUI

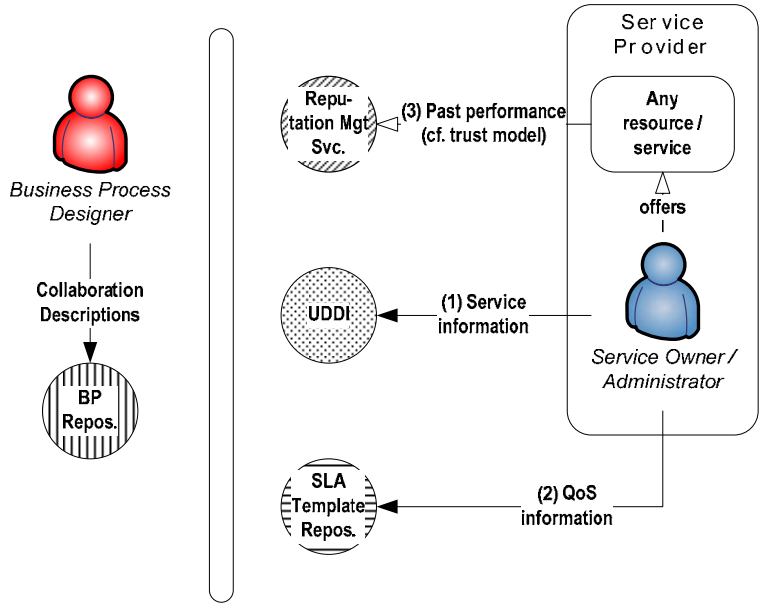


Figure 12: Components involved in the preparation processes.

We do assume that a service provider will publish its resources' functionalities (Figure 12.1) in some kind of Service Description Repository (e.g. UDDI). Note that "resources" in the sense here can range from abstract application services (cf. section I.4), supporting services (cf. section II.1.b) to complex aggregated "products" or even human beings. The description itself *should* provide detailed information about (a) the capabilities and (b) the interaction means for the respective resources – notably, these descriptions do not necessarily map to actual resource methods but may represent virtualised methods such as exposed by the Business Processing engine (cf. section III.1.d).

Such information will allow first of all that services are discovered according to the required functionalities (cf. section III.1.b) and that they can be integrated into a workflow (cf. sections III.1.c, III.1.d).

We do not make any prescriptions with respect to the way of defining this information - in particular description of the capabilities in a way meaningful for a computer, i.e. usable for automatic discovery. Notably, semantic extensions may be used for this purpose, but are not elaborated in the implementation work of TrustCoM as there is ongoing work by others to reach agreement on the best solutions to these problems whose outcome we have waited on for usage. Without loss of generality, this description could consist of a simple string-based role declaration identical for all services providing the same (or similar) functionality - though this is obviously not a very realistic assumption, it is still valid from a mere conceptual stance and is pursued as such in the implementation process (cf. D53).

Similarly, definition of the interaction means is required to allow automatic usage by other resources - whilst this is one of the main intentions of the WSDL specification, it nonetheless cannot fulfil this task alone: every service provider will specify his/her own WSDL thus exposing different methods and names, even though providing the same basic functionalities (i.e. roles). Accordingly, a client will have to identify the relevant methods first. Though this task may be supplemented by the service description, similar interpretation means are required to support full capabilities. Again, without loss of generality,

we may presume that all resources realizing the same role will provide identical WSDLs - such a simplification is conceptually valid assuming additional translation capabilities.

The service provider may furthermore want to expose information about the quality the resources may support (Figure 12.2) - as detailed in the following section, such information may be used in order to identify service providers that maintain specific quality parameters, like response time, but also offer the most appropriate pricing schemes etc.

Finally, it may be assumed that the service is registered at a reputation management service (Figure 12.3) and has already built up a reputation through past performances (cf. Appendix and section I.5) – though such information is helpful for identifying services on a trust-reputation basis (cf. below), this can not be considered a *requirement* as in particular SMEs and their services new to the eBusiness domain will not have earned a reputation and would hence be "invisible" to a Virtual Organisation that requires such information.

Multiple reputation management services may exist, thus providing different views on the resource's reliability (providing that it is registered). As will be discussed, it is up to the VO to decide which reputation service to use and/or integrate for further registration.

Without loss of generality, we furthermore assume that a business process designer has stored typical collaboration descriptions (cf. section I.4) in an accessible business process repository that allows users to get collaboration descriptions for a specific business goal (see also Appendix). Notably, this process can be easily replaced with addressing such a designer directly, the customer providing the description him-/herself or similar solutions.

III.1.b Identification

The identification phase is generally considered to be the first lifecycle phase of a Virtual Organization. From the TrustCoM perspective, identification starts with defining the business goal and ends with a list of (potential) VO members and a set of negotiated contracts.

Since a customer, respectively an initiator of a Virtual Organization does not necessarily maintain the qualifications for defining full collaboration descriptions (cf. section I.4), the TrustCoM approach supports "business agnostic" customers by introducing a three-step conversion technique as described in section I.4

1) Definition of the abstract business goal and its boundary conditions, like overall budget, time frame etc.
2) Conversion of the business goal into a (potentially complex) workflow-like collaboration description that carries information about (a) which roles and (b) which additional services are required, (c) what requirements each participant has to fulfil and (d) how these services have to interact.
3) The collaboration description defines interactions according to the abstraction layer declared by the respective Service Providers (namely "products", cf. sections I.4, II) that hence needs to be mapped to the actual business processes of these SPs.

The information required in order to derive a collaboration description from an abstract business goal is provided by a simple repository, an actual business process designer or some other means.

Note that beyond the participants described in the collaboration description, a Virtual Organization may require additional support from services outside the VO boundaries

(supporting services), as well as integrated ones for taking over additional tasks (trusted third parties). Such services may not be specified in the collaboration description, as they may relate to specific VO structures, rather than business goals - the Reputation services used for reliability measurement and the actual Enterprise Network SPs are chosen from are examples of such services. Each VO Management provider and/or customer *may* specify such additional requirements according to their own discretion.

The participant definition in the collaboration description should ideally be in a form flexible enough to allow identification of different role descriptions and integration of various interfaces - in such cases, more intelligent translation techniques, as e.g. envisaged by the semantic web community, can be exploited by integrating them into the discovery service. Furthermore, the collaboration description may need to be adapted, since each application service provider may in itself alter the overall VO requirements, by not fulfilling the task(s) exactly in the way proposed by the CD, by requiring outsourced or subcontracted support e.g. by additional TTP services etc.. Similarly the actual details of the individual participants will implicitly influence each other, as e.g. the time and budget constraints of the overall process need to be shared by, respectively distributed to all participants.

We consider contract negotiation as part of the Identification phase, since rejection of the contract (due to lacking resources or similar) will potentially lead to (re)identification of alternative providers. With acceptance of the Service Level Agreement a form of electronic contract (as part of the GVOA, section I.6) is generated and sent to the Service Provider as part of an invitation.

As such, the results of the Identification phase consists in a list of Service Providers that have all agreed to participate in the Virtual Organisation.

In Figure 13 all the components involved in realizing the Identification related processes are represented together with their dependencies. Note that the initiator of the VO also has the option of providing its own VO management components, as is e.g. the case with the CE testbed scenario. Without loss of generality, we may assume that the instantiation of a Virtual Organisation is initiated by a customer with a particular business request (“goal description”) that he/she passes to a VO Management Service provider.

In order to initiate a VO that meets the customer’s business request, as well as the according requirements (such as available budget), the so-called Collaboration Description (section I.4) needs to be generated. As this process requires a lot more intelligence than currently provided through automated means, we assume that this description is either provided by the customer itself or from some other third party - for implementation purposes we make the simplifying assumption that a kind of repository maintains a set of collaboration descriptions fulfilling different goals under various restrictions or that the according description is provided by the customer him-/herself.

Using this description, the VO Manager will generate a general VO Agreement (cf. section I.6) that incorporates the requirements from the collaboration description.

The actual identification of participants is realized with the help of a discovery service that queries common repositories - belonging to a specific Enterprise Network, well-known public repositories or even specified by the VO initiator - to retrieve a list of service providers matching the requirements up to a certain degree. Notably the results of this query may imply an adaptation of the collaboration description (described above) when not

fully inline with the requirements. We recommend identification of service(s) that fulfil the requirements with respect to (a) functionality, (b) quality of service and (c) reputation. The latter only if the service, respectively its provider has already built up a reputation, as noted in the previous section. It is up to VO Management and/or customer to decide whether the risk of integrating the respective service is worth taking regarding the role it has to play in the overall business execution taking into account any available reputation information. This decision may be delegated to the VO Management services if the reputation threshold is low.

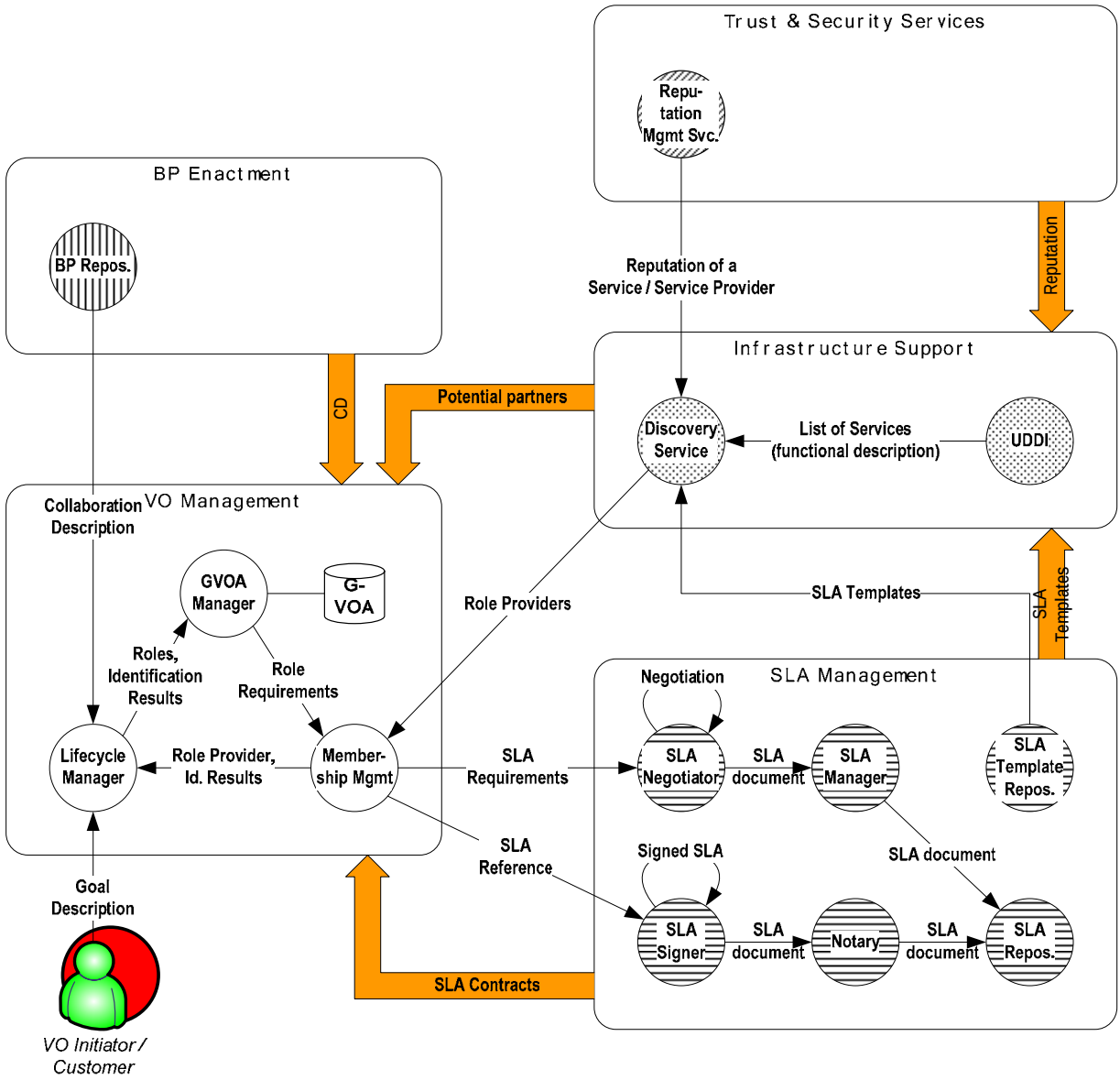


Figure 13: Relationships between components during identification

Notably, SLAs may involve more parties than just the service provider and consumer [19] like e.g. a trusted third party SLA Evaluator service (cf. Appendix) - even though these parties are not “signatory” partners of the SLA, their consent for providing the required functionalities and hence their availability is required so that they need to be involved in the negotiation process.

III.1.c Formation

As stated above we assume that a set of service providers is available by the end of the identification phase, that

(a) fulfil all roles of the collaboration description initially required, as well as additionally defined by the Virtual Organisation requirements or by the service providers (cf. Identification above)

(b) are available at the time needed in the form (quality of service) required, which has been ensured through the negotiation process.

Note that this may imply that some resources ("products") will be discovered at a later time during the enactment of the Virtual Organisation, as it is not sensible to occupy resources when not needed and may involve additional costs. Accordingly, the identification process may not result in a *full* set of services, so that identification and formation with respect to all other services will take place at another time, as specified by the collaboration description.

In order to allow for secure communication, as well as for monitoring, enactment of VO specific policies and to enable the distributed enactment of the collaboration definition, the services participating in the virtual organization need to be configured. The main task of the Formation phase is thus to prepare the operation of a VO in a way that allows for the overall (business) requirements, as stated in section I and in [2]. Notably one needs to distinguish here between (1) the configuration of the provided service (respectively the infrastructure) itself, e.g. so as to meet the agreed QoS, or to actually deploy the necessary services etc., and (2) the configuration of the components related to the underlying (TrustCoM) framework, e.g. providing the monitor with information what services to supervise how, deploying the policies etc. – whilst the former are related to actually providing the service, the latter cover the aspects with respect to that particular VO "instance".

As one would expect, configuration of the actual resources to be provided ((1) above) is completely up to the service owner, even though TrustCoM may support this process by triggering the right methods during the formation process, respectively by providing the means to adapt the (local) business processes As opposed to this, configuration of TrustCoM related components ((2) above) is mostly up to VO Management, though the service owner can (and partially will) have to provide additional configuration information, such as related to virtualisation of the resources. This is simply due to the fact that the information required for setting up local resources is private to the owner, whilst TrustCoM components are generally dependent on VO wide information. Note that a service owner replacing TrustCoM components with own implementations will either have to ensure that the respective components provide the same configuration interface or he/she will have to take of configuration him-/herself (cf. also chapter IV).

The Formation phase results in a fully configured Virtual Organisation that is ready for enactment - at least in so far as no additional services are required, as mentioned before. From now on, the service providers are actual registered members of the VO and as such liable for providing the service in the agreed upon form (cf. chapter I).

As can be seen from Figure 14, the main relationships during this phase involve distribution of VO specific information to each service involved, thus allowing the instantiation and

configuration of the respective components so as to meet the applicable requirements. Notably configuration and instantiation is strongly related to the EN/VO Infrastructure related components that are not depicted here for reasons of space (cf. above) – please refer to section III.2 for the respective details.

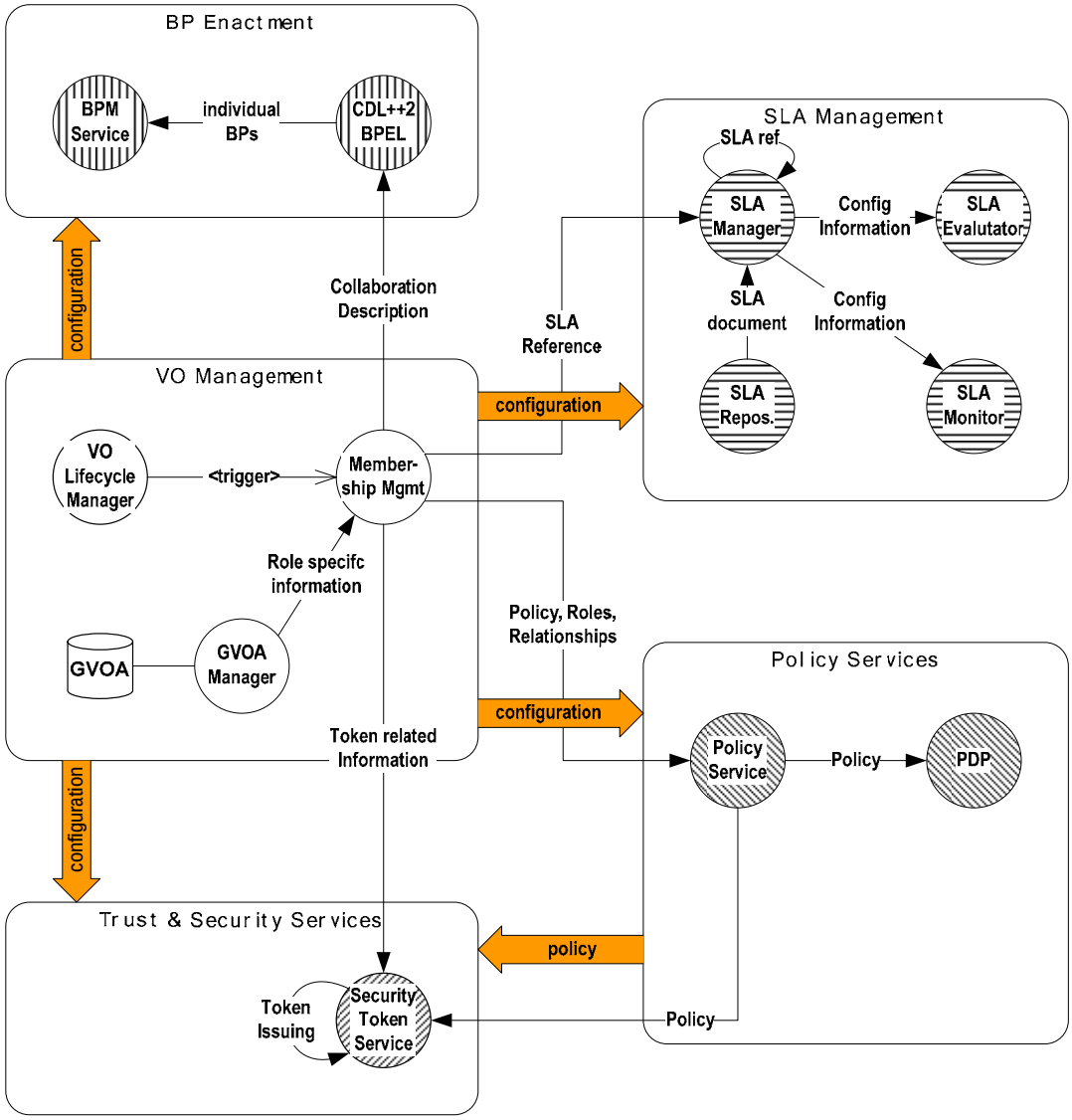


Figure 14: Relationships between components during formation

In order to realise automated support for Formation, each service must expose the relevant configuration capabilities. The components register for specific configuration information as described in the EN/VO Infrastructure section. In some cases a configuration capability is provided at the functional subsystem (Policy services, Security support, Trust-related services etc.) level. The sub-system configuration capability then takes over responsibility for passing the necessary data to all involved components, respectively instantiating them – a particular example of this approach can be found in the SLA Management subsystem, that uses one component (the SLA Manager) for most interactions with other logical systems (see Appendix for details).

It has to be noted that the Formation phase is not a single definite phase, in the sense that the processes involved will only be invoked once within the lifetime of the VO - rather just like with Identification, configuration of services may be required at various stages in the VO lifecycle:

(1) as the means to set up the VO, i.e. the second main phase of the VO.

(2) when service providers are integrated into the VO at a later stage, independently of whether they have been identified during the Identification phase – e.g. when a specific service is only required for a limited time and hence does not need to be configured for the whole lifetime of the VO. In such a case, the actual integration time has to be specified in some way in the collaboration description.

(3) when service providers are replaced dynamically at runtime by other providers, i.e. involving additional identification processes, too – this is detailed in section III.1.e.

III.1.d Operation

Once a Virtual Organisation is set up according to the requirements derived from some overall (business) goal, the participants may principally start cooperation in a way correlating with the general collaboration description. Such enactment will consist in successive invocations of the individual application services, i.e. the passing and processing of data sets between each other. Business Processes realized by such Virtual Organisations are not restricted to simple data processing, as the actual roles to be fulfilled by the individual participants may be front-ends to any complicated tasks, involving human beings and any type of resources, that however communicate with other participating entities through the means of web service based message exchange.

In accordance with what has been stated in section I.4, one has to distinguish between the VO's view on the business process and the view of the individual participants: whilst the former focus on the message exchange between the service providers, but does not provide any details regarding the actual execution of the individual roles, the latter describes the details per role and intermediate interaction partners, but does (in itself) not allow insight into the overall process.

A straight-forward approach would hence foresee a central "business process engine" that triggers the actors (of the overall collaboration) corresponding to the pre-defined sequence and forwards the respective data sets accordingly. However, such an approach produces a bottleneck in messaging, would cause unnecessary delays, in particular with huge amount of data, and introduces a single-point-of-failure. Each participant in a virtual organisation will have been provided with his/her role specific information of the collaboration description during the formation phase that each participant can turn into applicable ("internal") business processes (cf. Appendix B and sections III.1.a, III.1.c). According to the definition of the collaboration description, these role-specific parts will already contain the relevant contact information, i.e. data source and destination.

These contact details do not specify the identity of a particular partner or service instance directly, but must provide enough information for the message to be routed to the correct destination. The process of identifying the interaction partner is described in more detail in the sections III.1.a, III.1.b, as well as in the Appendix, sections I.1 and I.6.d. Note furthermore that the contact information may change during the execution of the respective

provider's service, which will require updating of the details – see the description of the Evolution phase below.

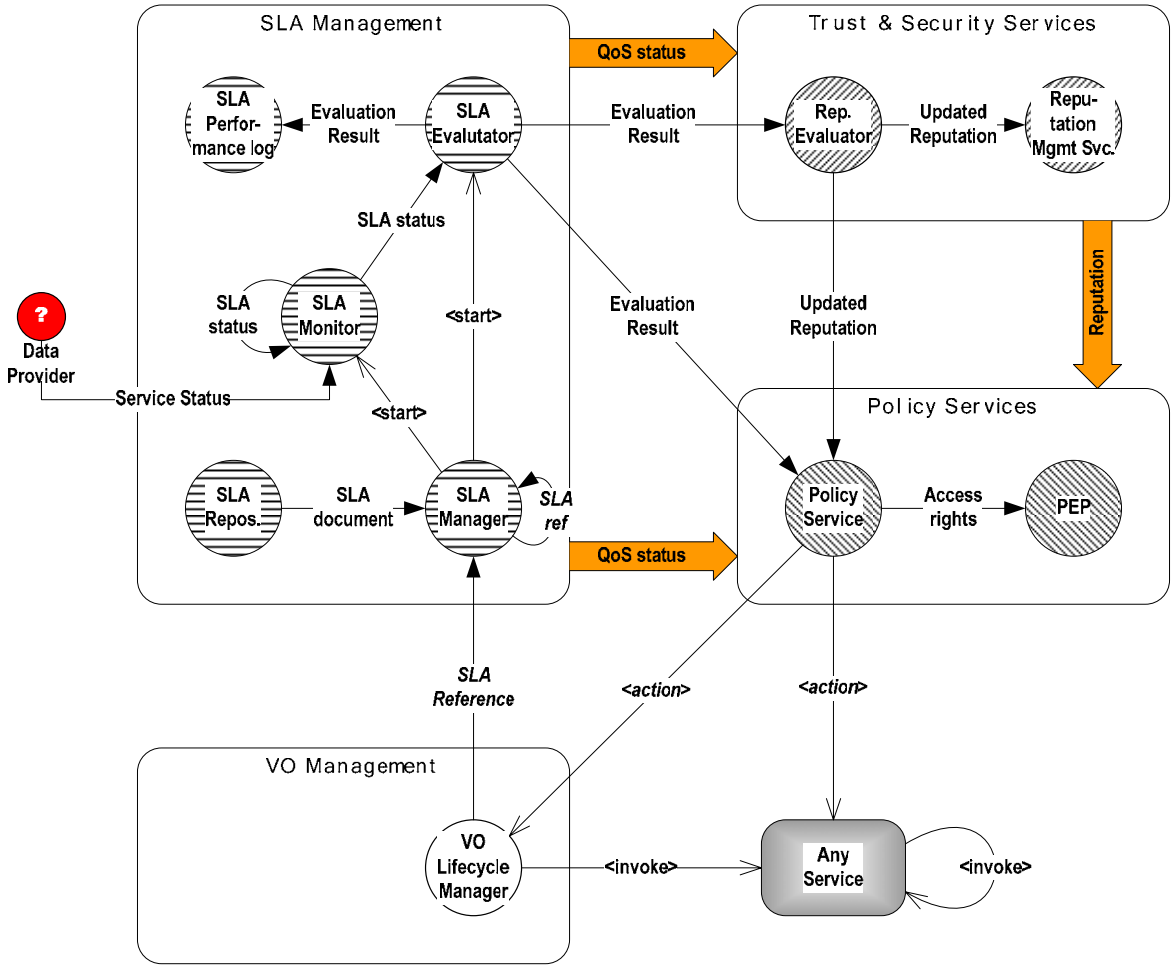


Figure 15: Relationships between components during operation

Besides for the relationships between the enacting participants, i.e. the application services and the involved supporting services, respectively trusted third parties, QoS monitoring plays an important role during the Operation of a virtual organisation (cf. Figure 15):

Services that are subject to QoS terms will be constantly supervised during their enactment with regards to the SLAs agreed upon during the Identification of the respective service(s) (cf. sections I.6, III.1.b and Appendix B, section III). We assume that the respective information is gathered through so-called "data providers", whereas it is of no importance for the framework where they are deployed - however, the *type* of information required will have a direct influence on the localisation of the respective tools (cf. chapter IV). The current status of the service provider with respect to the negotiated SLA may be distributed to different interested parties – besides for the customer, in particular the policy and reputation related services that require this information e.g. for taking SLA related actions (cf. Evolution, section III.1.e).

SLA related performance information is in particular of relevance for measuring the reliability of a Service Provider – since such information does not directly map to scoring, an intermediary "evaluator" needs to convert the data according to the Reputation

Manager's business logic (scoring basis). At the same time, this serves as a means to "neutralise" reputation information, so as to meet confidentiality issues with respect to such data (cf. section I.5).

The main basis for defining and enforcing decisions in a Virtual Organisation – not only for triggering the evolution actions on basis of SLA violations, but also for specifying the access right restrictions (cf. EN/VO Infrastructure description, section III.2) and describing the general event-condition-actions in a VO (cf. Appendix, section I.5) – consist in policies. These will mostly cover "unexpected" events in the VO that are not foreseen in the collaboration description, and as such generally relate to Evolution processes (cf. section III.1.e). In relationship with the overall collaboration definition, policies may also be exploited to steer the overall progress depending on environment conditions, like e.g. changes in the market demand – however, without loss of generality, this may be considered Evolution, since it entails the reconfiguration of the Virtual Organisation (see below).

III.1.e Evolution

Within its lifetime, the participants and configuration of a Virtual Organisation will most likely be subject to multiple changes, i.e. service providers may be replaced, security settings altered, the business goal redefined etc. Though this is part of any "normal" operation of a VO, we consider it a (sub)phase of its own as it will generally lead to partial repetition of Identification, Formation and Dissolution processes.

The actual causes invoking Evolution are various and may actually change between different VOs, as they may be (co)defined by the initiator and the Collaboration Description. Besides for the individual ones, some common triggers may be identified that are recommended to be considered in a Virtual Organisation:

- SLA Violations

  Generally, violating the SLA contracts by not meeting specific QoS related parameters, or – more generally – by not providing the performance as agreed upon during negotiation, will lead to some form of compensation to be provided by the violating party, like paying a fine. However, repeated violations or severe "contract" breaches may lead to complete replacement of the respective service/provider, which implies dissolution (for the specific partner), potential re-identification of service providers (in case the alternatives were not maintained during the initial Identification phase), new negotiation and re-formation.

  Whether that member will actually *be* replaced depends on a number of factors relating to the overall goals of the VO – as such, e.g. low time-constraints may be a relevant factor for maintaining even mal-performing parties, since a replacement may delay the overall process too much. In all cases, *availability* of alternative providers will play an important role.

- Reputation drop

  Since business entities will provide their services to more than one customer (or here the virtual organisation), their performance in different relationships will feed back on their reputation (given that they are registered at some reputation management service in that respective business relationship). Accordingly, the TrustCoM VO will analyse

updates in the participating parties' reputation and take according actions once the reputation drops lower than the overall requirements allow. Low reputation of a provider implies an increased risk of their misbehaving with respect to performance, security issues and similar aspects, depending on the "type" of reputation (cf. section I.5) and should hence be circumvent by the virtual organisation as best as possible – this may imply increasing the security thresholds, lowering access rights, etc. up to the point of replacing the service (cf. SLA violations above).

- Changing Location / EPR

Resources may change on behalf of the service provider, e.g. by changing the address of the respective machine, by moving the resource to a different machine etc. Such changes generally imply modifications of the contact specific information alone (see EN/VO Infrastructure, section III.2) – however, more radical resource changes, e.g. moving a resource to a non-EC state may imply changes on the security aspects of the VO. We must assume that such restrictions, respectively consequences are defined in the GVOA and hence the VO Policies.

- Non-responding Participants

Participants, in particular services that fail to respond within a given time to VO specific requests (invocations) need to be considered unavailable so as not to delay the overall processing of the VO for too long. The actual timeout delay will vary between individual VOs and even between participants, depending on how time critical the provided service is – as such, e.g. a frequently used calculation web service may be more time-critical than a simple file backup service.

Non-responding parties will generally have to be replaced, as it must be assumed that the service is down for good and hence can not take over the respective task(s) again.

- Unassigned Roles

It may be the case that a role of the collaboration description is not assigned right from the beginning, since it is not required for the whole duration of a Virtual Organisation. In such a case, the (potential) Identification and integration (Formation) of the role provider is considered Evolution.

- Lacking Role Providers

As the identification attempts (when replacing a member or when assigning a role during operation) may not necessarily lead to actual results, i.e. if no suitable service provider for a specific role can be identified, the collaboration may have to be reconfigured completely. This may range from re-negotiation of individual terms up to designing a new collaboration description – this issue is discussed up to some degree in the section on Identification (III.1.b) above.

- Security Violations

Repeated Intrusion attempts, like repeated unsuccessful authentication or endeavours to access restricted resources, may indicate severe attempts to breach the security of the Virtual Organisation. Thus such attempts will require a reconfiguration increasing the security thresholds, in particular logging of the invocations and their sources, and may possibly even result in changing the contact points to hinder further attempts. As such efforts may also be initiated from *within* the VO ("malperforming" partners),

counter-measurements, in particular fines and potential removal of the member, need to be enacted.

- Changing Environmental Conditions / Customer Request

Since the Virtual Organisation is created to meet a specific, potentially temporary business objective, such as covering a market niche, changes in the environment (e.g. market niche being sufficiently covered by other enterprises) may lead to externally triggered reconfiguration of the VO (either by customer, VO Management or specific VO policies designed for such occasions). This does not necessarily imply termination of the virtual organisation, as it may be possible to compensate for the changes by adapting the collaboration description, e.g. to fill a similar, yet less covered market niche.

Notably, expulsion of a VO member may trigger a legal process when the service owner objects to being removed from the Virtual Organisation - whilst TrustCoM can not take charge of these legal actions, as this requires the capabilities of a human lawyer, the framework can support the tasks by providing performance history and related information. Due to legal restrictions[19] any participant that is replaced in the VO needs to be informed of this action first – implicitly, processing may stall until the replacement is acknowledged by the respective provider. Such a delay may prove hazardous to time-critical operations (like e.g. in the AS testbed) and should be kept to a minimum – in such cases it may be advisable to integrate a substitute Service Provider that takes over the according tasks at least as long as the potential litigation lasts.

The actual behaviour depends very much on the underlying contracts and the actual business goals of the VO – as such they may not be fully stated in the VO's policies but decided on case-by-case basis.

Even with a dynamic system such as TrustCoM, evolution and thus changes in the structure and/or configuration of a Virtual Organisation will *always* bear a high risk of delays and even of failure of the business goals, as the following risks may arise

(1) no alternative service provider is available to take over the (missing) role which leads either to complete restructuring of the overall collaboration description or even to failure of the VO

(2) the alternative service provider does not meet the role specific requirements so that either the overall requirements have to be adapted or the business goal can not be met in its current form

(3) the changes resulting from the replacement provider lead to changed conditions for other participants which is not accepted and may lead to renegotiation or failure

In any case, such aspects need to be taken into account when writing the event-condition-action rules that determine under what circumstances a service should be replaced, or "milder" consequences will be taken (see also Identification, section III.1.b).

[19] DIRECTIVE 2000/31/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 8 June 2000 on certain legal aspects of information society services, in particular electronic commerce, in the Internal Market (Directive on electronic commerce).

Figure 16 shows how a policy induced replacement of a specific member relates to other processes in the Virtual Organisation – though focussing here on replacement, the relationships would be similar for a reconfiguration of the participants, yet without making use of the additional steps for actually expulsing the member.

In particular with respect to “severe” measures, like dispatching a specific party due to SLA violations, i.e. even though the service is still existent, TrustCoM needs to take potential errors on behalf of the VO services into account - this includes failures on behalf of the monitoring and evaluation components etc. The SLA performance log, as well as the reference SLA document stored at the Notary will provide additional information for identifying the source and “justification” of the replacement action. During this process, execution with respect to the party in question needs to be interrupted to avoid failure and reduce the risk of misbehaviour of e.g. “doubtful” participants, i.e. when the respective reputation has dropped below a critical threshold. Notice though, that the legal requirements make all further actions impossible until the SP owner has been fully informed about this process. Once the legal actions have been taken, access to other participants will be restricted (at least temporarily) to avoid potential misuse, except for resources that keep information of direct impact to the respective Service Provider, like e.g. performance logs.

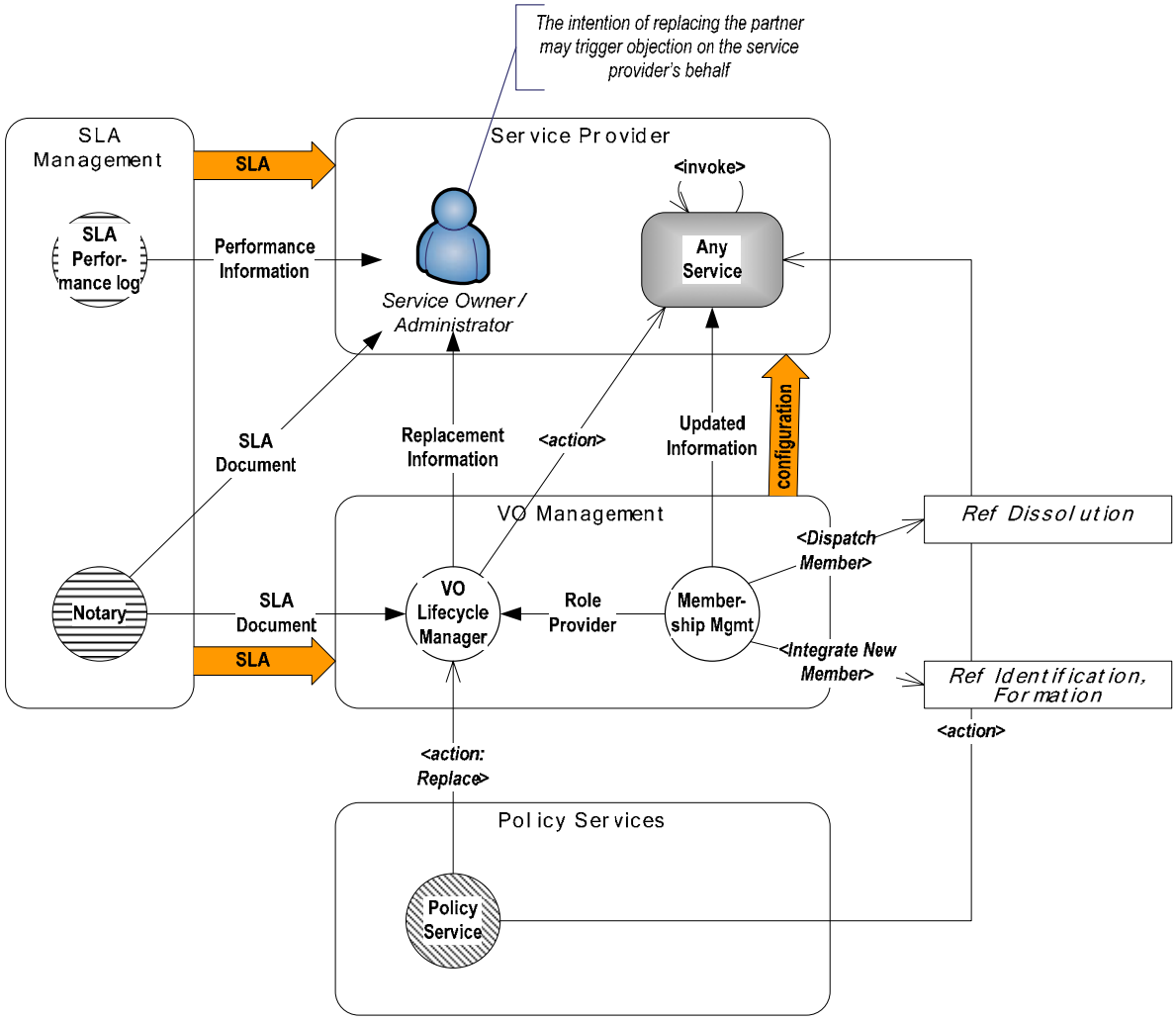


Figure 16: Relationships between components, services and service providers during Evolution.

As can be seen from the figure, the relationships are basically identical to the ones of the Identification, Formation and Dissolution phases of the Virtual Organisation (described in the according sections), whereas the respective processes only differ with respect to how many services are affected by the Evolution: whilst Dissolution will only affect the party to be dispatched (in case of replacement), Identification will also be involved when a non-assigned role needs to be manned or contracts need to be re-negotiated. Finally Formation includes all the necessary reconfiguration steps that involve in most cases *all* participants for adjusting access information, providing updated security tokens etc. (cf. section III.2).

With successful reconfiguration, enactment of the overall collaboration may continue, though potential “rollbacks” need to be taken into account when the execution-state by the respective provider gets lost, respectively can not be taken over, or the execution can otherwise not simply continue from the time of interruption, e.g. due to slight changes in the means of generating the data between the new and the replaced service (cf. appendix, section I.2)

III.1.f Dissolution

The Dissolution phase marks the end of the VO lifecycle, though not necessarily for the whole Virtual Organisation, but potentially only for individual participants that are not required anymore or that will be dispatched, respectively replaced due to violations or similar issues (cf. Evolution above). The two processes differ only slightly since dissolution of the whole VO is conceptually similar to dispatching all its members. We do assume here that in either case the respective legal conditions have been fulfilled, i.e. that the contract either has come to its predefined end and/or the SP owner has been informed of the according steps (cf. section III.1.e).

For each member to be dispatched the respective execution needs to be halted and, in particular for participants with low trustworthiness, their capabilities of accessing other services and/or data needs to be restricted, so as to reduce the risk of the respective entity inflicting potential damage upon other participants in the Virtual Organizations, or even the whole execution. This implies that all other members are informed of the respective changes in time to avoid problems with executing the process when interactions with (to be) dispatched entities are required.

Dispatched members will also want to go through the process of auditing, where it is ensured that the entities will receive payment for the service they have provided. In addition to this, we consider reputation (and thus trustworthiness) of participants with respect to their performance in VOs an important issue for supplementing security aspects and reducing the overall risks of execution failure – thus auditing for TrustCoM may involve assessment of the respective providers’ performance with respect to SLAs and other policy violations, insofar as they are monitored by the Virtual Organization (cf. section III.1.d). Also refer to section I.6 for details about dissolution with respect to the VO contract.

Functionally, the Dissolution stops all active business processes of the according service(s) and destroys all security tokens and policies that implicitly define the access rights of the respective service – revocation of such access rights here means that all participants in the virtual organisation are instructed not to accept the respective tokens any more.

Furthermore, the SLA contracts with that respective service(s) are annulled and all SLA management related services stopped, since the monitored data is no longer valid and would cause unjustified violation messages.

Most companies provide their own financial auditing services upon which they rely due to contractual reasons and it can be generally assumed that they will not want to switch to a different system / Service Provider. Accordingly, TrustCoM does not provide new means for auditing but supports means of exploiting the SLA Management capabilities for such purposes either by subscribing to the respective notifications or by reading the SLA Performance log. Actual payment details (costs, fines etc.) are specified within the Service Level Agreement and have to be used for auditing purposes.

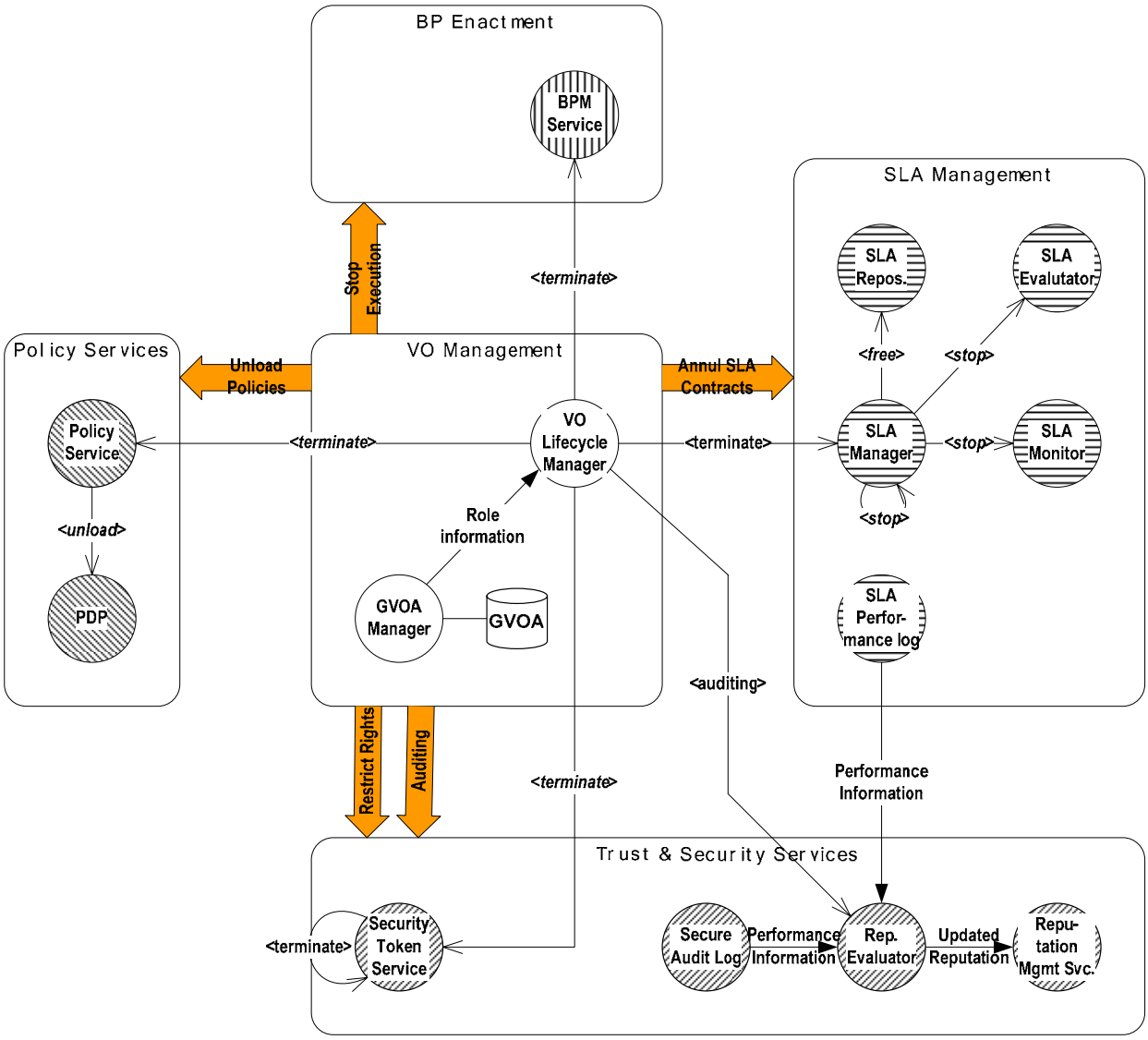


Figure 17: Relationships between components during Dissolution.

With respect to auditing trustworthiness related parameters, the VO-“local” Reputation Evaluators convert the SLA performance information (either from log or during operation by according notification subscription) into into trust values meaningful for the more global

Reputation “repositories” that can be accessed by other interested parties in the Enterprise Network or through other means (cf. sections III.1.a, III.1.b, see also section III.1.d).

Notably, any changes to the configuration will imply changes on all members *involved* with the respective party, as restricting access rights, revoking tokens etc. really requires reconfiguration of all related entities, as detailed below in the EN/VO Infrastructure section. All changes are maintained in the GVOA.

III.2 The Relationships in the underlying EN/VO Infrastructure

The EN/VO Infrastructure related processes behave somewhat differently from the components described in the preceding sections, since they build the underlying basis for uniform messaging, accessibility and coordination of interactions. As such, they participate in some way in most interactions but are largely invisible to the service providers. Their roles include support for interaction across enterprise borders (notifications, messaging, logging), as well as deployment and management of service and component instances (service instantiator, service instance registry). Notably discovery support (discovery services and additional repositories) has already been discussed in section III.1.b of this document, due to its strong usage in that phase and this will not be repeated here, even though these functionalities are grouped as belonging to the EN/VO Infrastructure (cf. section II.1.b).

From a functional point of view the EN/VO Infrastructure extends the (virtual) services exposed by the service providers with VO capabilities that will allow the entity to make use of the functionalities summarised above. To realise these extensions in a virtual organisation, the respective counterparts in VO Management level are required, so that the diagrams below give no indication of the component distribution across participants and management instances – for such information refer to chapter IV, respectively to Appendix B.

Though the EN/VO Infrastructure follows the overall VO lifecycle, we distinguish here only 3 phases, namely Setup, Messaging and Evolution since the actual usage of the related components overlap with all phases. As such, e.g. the messaging capabilities described below will already be partially used during Formation, whilst Evolution captures also aspects of Dissolution. This way we avoid repeating functionalities in different diagrams – since the phases on this level are generally not explicitly triggered but implicitly invoked through processes on the service level, this overlap of phases does not cause functional problems.

Virtualisation

One of the main functionalities covered by the EN/VO Infrastructure support consists in “virtualising” the Service Provider’s capabilities inside the Virtual Organisation. This means in particular that the gateway allows exposing methods and functions that *as such* do not link to actually existing resources and/or their methods. Rather, the interface allows for mapping of invocations to (intermediary) services’ functions that e.g. trigger complex workflows in order to realise the functionalities as published by the Service Provider.

This functionality enables Service Providers to expose functionalities that comply with their product lifecycle rather than with the resources they maintain. From the VO perspective, the gateway exposes a (virtual) resource that can be used via normal Web Service

invocations. This complies with the “abstract entity” approach as described in section I.4 and chapter II.

III.2.a Setup (Formation)

Most services in a Virtual Organisation need to be stateful or at least individually configured for the respective requirements, e.g. when the service is subject to QoS parameters. As such, these services will require instantiation and configuration before they can be used – though the instantiation details will be different between service providers and may even be private. Relevant configuration details hence need to be distributed to each Service Provider and at least a trigger-like indicator has to be given when the instances are required to be setup. Also, TrustCoM specific components implicitly follow the instantiation procedures.

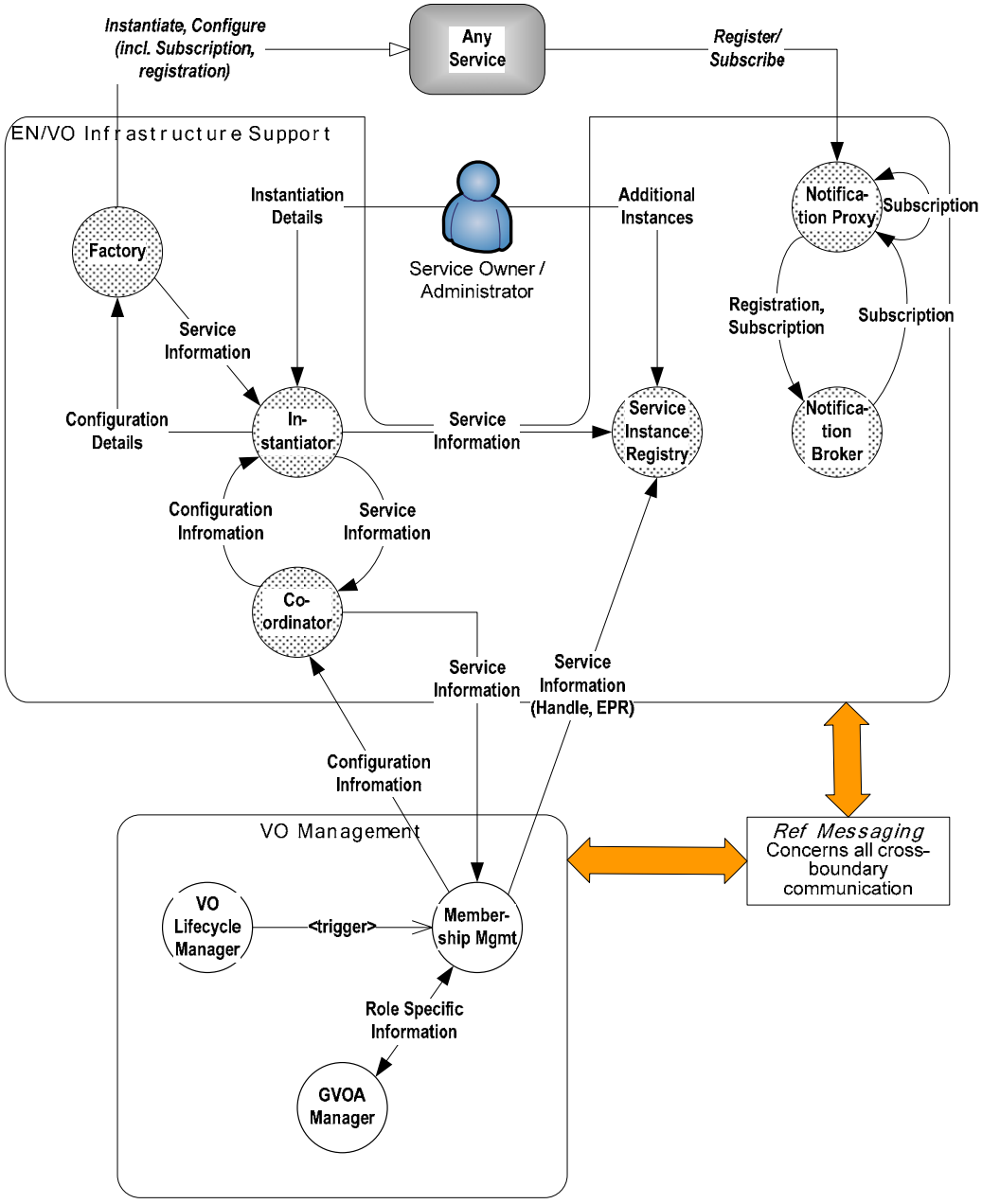


Figure 18: EN/VO Infrastructure interactions to set up a VO.

Since these instances may change during enactment of the virtual organisation, generally due to some Evolution processes (cf. section III.1.e), it is not sensible for services to interact *primarily* with exact Endpoint References (short EPR) since every change would require updating all related information, like e.g. the business process. In order to avoid this problem, TrustCoM deals with individual interaction partners on the basis of *handles* that are resolved by the message interceptor (cf. Messaging below).

Besides for direct interaction, much information distribution during operation of the VO takes place as topic-based notifications, thus informing (a set of) interested parties of specific events that take place in the Virtual Organisation. Types of events are distinguished by “topics” in order to reduce amount of messaging and to allow subscribers to pre-select only those events that are of interest to them. Even for notifications we see

the requirement of confidentiality, so that some messages may not be received by all participants.

In Figure 18 we depict the processes, respectively relationships that partake in setting up the components for these functionalities: the instantiation process and according distribution of configuration details proceeds in a coordinated way so as to avoid that instances are required before they have been instantiated. The detailed information of these instances is provided to the so-called “Service Instance Registry” which takes over responsibility for resolving the aforementioned service handlers (see Messaging below for details).

Configuration details and potentially additional information from the administrator will furthermore convey details with respect to what notifications need to be provided, respectively received by the individual services, thus triggering the subscription and registration processes at the notification related components. As such, a service provider may add individual, local endpoints to the service instance registry, thus redirecting specific invocation calls to (only locally known) service instances or even for enhancing messaging between local messages (cf. below).

III.2.b Messaging (mostly Operation)

In order to enact the functionalities of TrustCoM upon interactions between participants, respectively services in the Virtual Organisation, the messages need to be converted (in order to meet the VO messaging requirements, like signature, encryption etc.) and verified accordingly. In theory, all outgoing messages of the actual service (be it application service, TTP or VO Management), should be enhanced in a way that allows uniform understanding within the VO, including identification of the actual endpoint from the handler (cf. above). To complement this, all ingoing messages should be verified with respect to access rights and authentication of the sender.

To realise these capabilities, all participants need to support some kind of message enhancement / verification system as a kind of “front-end” or gateway to the actual service(s). This gateway acts as the actual contact point for interacting with the local services, thus allowing local redirection of messages that is not visible from outside of the respective service’s domain, as well as global redirection to endpoints according to some kind of endpoint identifier, like e.g. the rolename. Whilst the former ensures that the right resources (including workflow engines etc) are reachable even in a confidential, private infrastructure, the latter allows communication across the Virtual Organisation without having to change the resources so as to address the right endpoints (in particular with respect to dynamicity).

As can be seen from Figure 19 and Figure 20, the processes behind message reception and message sending are principally identical, even though the purpose of these mechanisms alters slightly:

Sending Messages

As mentioned, sent messages should principally be extended by the VO specific requirements thus allowing uniform interactions. An additional focus rests on resolving the service handlers that are actually used for sending to valid endpoint references.

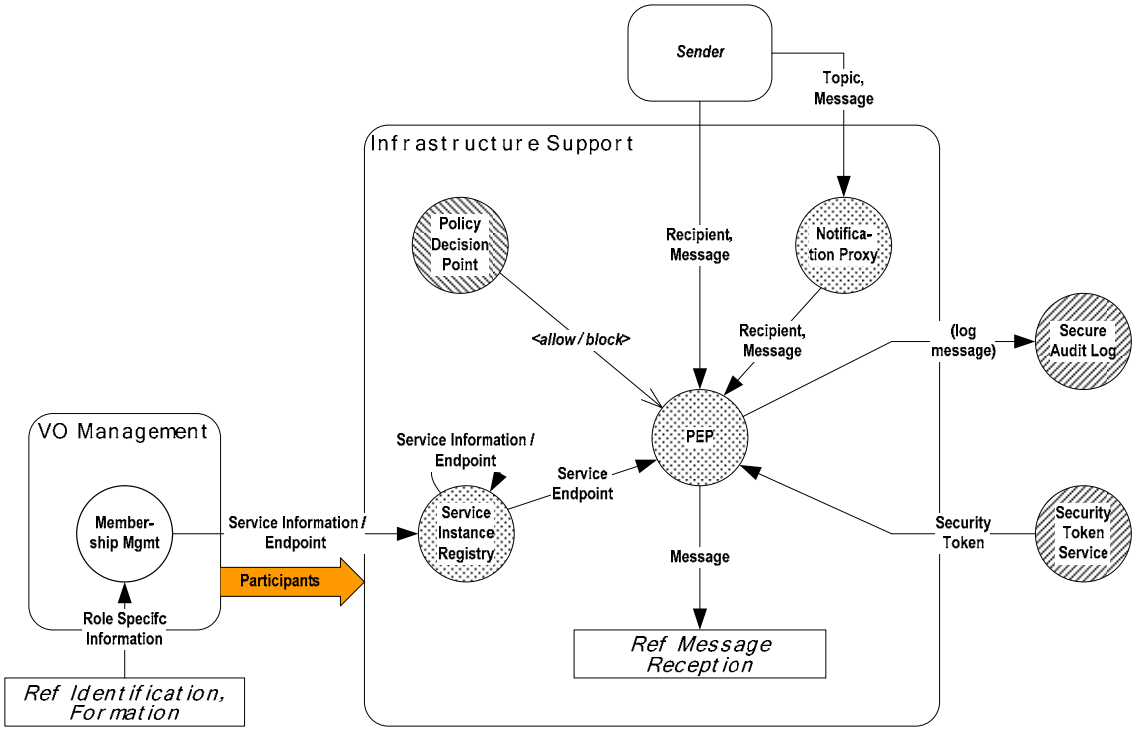


Figure 19: Processes involved in *sending* messages.

As Figure 19 shows, XML documents may either be sent as notifications or as “direct” Web Service invocations – whilst there is no actual technical difference between the two (both being SOAP messages), the former is realised indirectly through the means of some Notification Proxy interface and complies to a specific notification message standard. The notification support will maintain a list of all interested parties that potentially receive the message (given that they have been subscribed during Setup as described above).

All messages may now (1) be verified for whether they are allowed to be shipped to the recipient and (2) be extended by a security token to authenticate the sender within the VO – note that the Security Token Service and the Policy Enforcement Point (PEP) are described in more detail in chapter III.1.

The actual Endpoint References (EPR) for individual recipients are provided by the Service Instance Registry, which may need to query the VO Membership Management if the respective handle is unknown. This may be due to the fact that the respective participant has not yet been assigned and hence no instance or contact point exists for it – in such a case, trying to access it would need to interrupt the process and trigger Identification and Formation of that respective role.

Notably the identified EPR may not be the one of the actual recipient, if message brokering is desired for hiding the true identity of a service from either recipient or sender, e.g. if the sender of a message is not allowed to know the true location of the recipient for privacy reasons. In such a case the recipient as detailed below will consist of an intermediary service that acts as a Broker forwarding the message to the desired endpoint, potentially eliminating information about the sender by replacing the EPR with the handler again.

Receiving Messages

The recipient of a message, be it Broker or actual destination point, will want to ensure that the message was sent by an actual member of the Virtual Organisation and that he/she has the right to access the resource provided by it, thus minimizing the risk of data theft and other potential misuses.

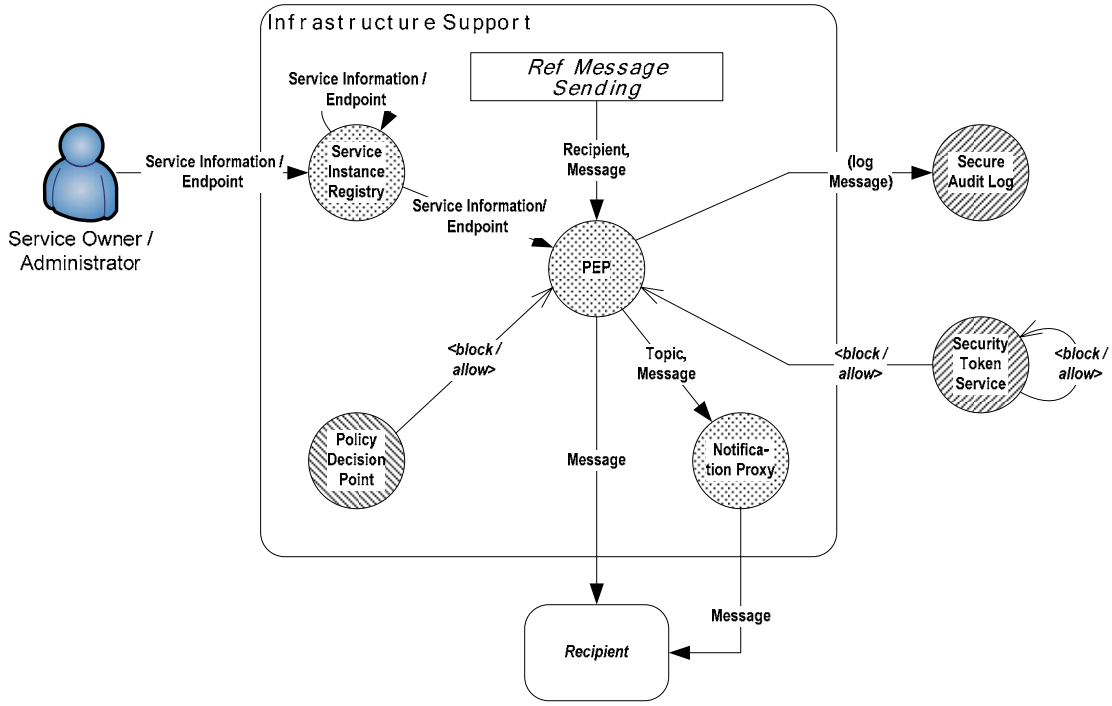


Figure 20: Processes involved in *receiving* messages.

As such, Policy Enforcement Point and Security Token Service will be queried to (1) authenticate the sender and (2) verify its access rights, potentially leading to the message being dismissed and a potential security violation being logged for later reference.

It has already been noted that the front-end as provided by the Message Interceptor does not necessarily reflect the real structure of the participant's domain – thus the actual recipient may need to be identified first on basis of the additional instance information provided by the service administrator (cf. Setup above). Again, the actual EPR may be missing, which in this case requires the interaction of the Service Administrator, since the domain-internal structure is principally unknown to the VO. Note that in case of message brokering the message would leave the domain again, thus starting the processes for sending messages as describe above again. Note also that notification messages may be distributed to different Endpoints within the domain, thus requiring the Notification Proxy as the actual destination of the message.

III.2.c Reconfiguration (Evolution)

With replacing or just dispatching a participant in a Virtual Organisation, it has to be ensured that the respective entity can no longer access resources in the VO, so as to avoid potential misuse of data (cf. discussion in section III.1.e). This implies not only that this specific service provider can no longer query resources, but also that other providers do not forward information, e.g. as part of a business process to that entity. Note that this does

not apply to *all* relationships inside a Virtual Organisation, as specific resources still need to be accessible to the Service Provider due to legal restrictions, as discussed e.g. in section III.1.e.

Since the evolution process may involve several steps, including potential objections on behalf of the member to be dispatched (section III.1.e), preliminary restrictions need to be activated immediately, as the overlap must be considered a potential security threat – these restrictions may be deactivated again, once it turns out that the Evolution procedure is not valid.

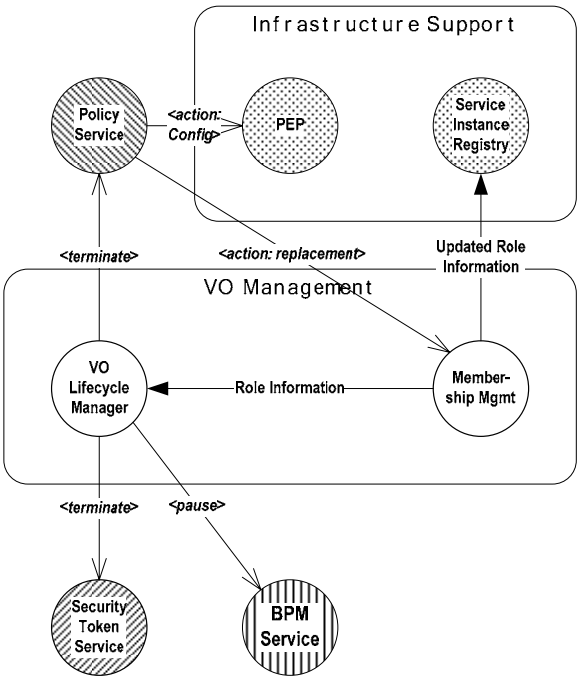


Figure 21: Evolution, respectively dissolution relationships in EN/VO Infrastructure components.

Most of the required processes take place on service level (cf. sections III.1.e, III.1.f), like revoking security tokens, unloading policies and interrupting the execution of the business process – this implicitly blocks all messaging attempts due to the lacking access rights and invalid authentication tokens. In addition to this, the contact information for that respective service provider will be removed to render all information passing *to* that entity impossible.

If the entity will be replaced rather than dispatched, the updated contact details will replace this obsolete data during the repeated Setup phase, as described above.

IV Deployment Model

This chapter depicts the consortium's recommendations with respect to deploying the TrustCoM framework for different use types (respectively business models). As opposed to previous versions of this document, the section does no longer provide detailed deployment examples since these could not reflect the whole range of deployment possibilities: with the strong Service Oriented Architecture approach of TrustCoM, as well as its main focus on Web Service based components, the framework allows *principally* any deployment, provided that the communication links are maintained as described by the relationship view (chapter III) and/or in Appendix B to this document.

The chapter is divided into two main sections:

(1) the general discussion which provides an overview over the general issues and considerations to be taken into account when deploying the TrustCoM framework

(2) the business model related description in which the recommendations and particular issues with respect to deploying the business models as described in section I.2 are detailed

Typical sample deployments for specific business use cases will be discussed in more detail with respect to TrustCoM's two testbed scenarios – please refer to chapter V.

IV.1 General Discussion

Within this section the minimal requirements to allow web service based resource provisioning will be indicated and some of the most typical deployment structures discussed - these depend on (a) the requirements of the service provider regarding TrustCoM support, (b) the type of service to be provided and (c) the existing infrastructure. The description applies to VO Management, Application Services, Trusted Third Parties and Supporting Services as well as to application-specific services provided by the VO partners.

As opposed to version 3 of this document, this section no longer provides a fully detailed deployment for each type of infrastructure, as the Business Model Specific Deployments section below, combined with the discussion in this chapter addresses this information implicitly. For detailed infrastructure deployment models, please refer to D62 (TrustCoM Framework V3).

IV.1.a General Requirements

TrustCoM addresses resources as Web Services with Grid extensions, thus any business entity wanting to provide services to a TrustCoM VO will require either a Web Server to host the gateway structure and some connection between this server and the actual resources (independently of whether they are hosted on the same machine or connected by intra-/internet), or at least that the actual resources are exposed as web services – note that this does *not* imply that the actual resource is a web service in itself, but rather that it is accessible via a web service interface. People may participate as resources in a VO if an appropriate 'user agent' front end is provided.

Supporting Services

Supporting parties are “outside” of the Virtual Organisation and as such do not experience any alterations through TrustCoM – in fact one of the most important issues to observe from a technical stance is that the existing and recommended services (mostly repositories) can be used. TrustCoM caters for this through the use of web service standards and respective recommendations.

Trusted Third Party

Though Trusted Third Parties are in the first instance quite “generic” services (logs, notary, notification broker), they nonetheless are VO specific: as discussed above (cf. section II.1.a), TTP services are considered “trusted” since they may have to maintain sensitive data for the participants and keep them private. Accordingly TTP services need to observe at least the security rules and the access rights of the Virtual Organisation.

Since TTPs may interact with other services for querying data or triggering actions, they will also have to be provided with the necessary contact information that allows identification of the EPRs of the respective interaction partners, or at least usage of message brokers to convey the relevant documents. Note that Notification is also a valid form of interaction for Trusted Third Parties. Being stateful themselves, at least with respect to maintaining the configuration parameters of the VO, these services furthermore need to provide some form of instantiation and support the means for registering the according instances in the virtual organisation.

TTP services may be, but generally are not subject to Service Level Agreements, since the *quality* of the resources is not adapted to the VO’s needs (as opposed to the service itself, which needs to provide the security support etc., as mentioned). Note that this does not imply that the service provider is not subject to a (legal) contract, but only that constant monitoring of the performance as such is not required and that generally no consequences from lacking performance arise.

Furthermore, it may be claimed without loss of generality that Trusted Third Party Services provide *atomic* functions, i.e. no aggregation of individual steps e.g. by execution of a business process. This implies that individual adaptation of the resource itself according to the business requirements of the Virtual Organisation does not impact on a workflow engine, thus eliminating the need for business process support.

Obviously, the two assumptions - no SLA, no Business Process - do not hold true for all use cases and may be regarded just as a “basis” configuration. In any other case, the service may be set up like an Application Service (see below).

Application Services

Application Services are the main contributors to a VO’s business goal(s) and as such require the most complete configuration and setup, thus realising the requirements identified by TrustCoM (cf. chapter I). An Application Service as provided to a Virtual Organisation consists in principle of any number of resources, services and (human) workers that are aggregated and directed by a business process to expose a specific functionality (the “role”). The individual infrastructures, distribution of tasks, as well as the business process details are principally completely up to the service provider, as long as they comply with the overall VO requirements. The aggregated functionality is exposed as

a single (web) service with no information about how the exposed methods map to internal processes.

Like Trusted Third Parties, Application Services need to be configured to the respective Virtual Organisation so as to ensure data protection, access restriction and common messaging formats. Accordingly, the base structure of the two service types and hence the base requirements are identical, up to the point of Service Level Agreements and Business Process execution:

In general, we must assume that Application Services consist of complex resource aggregations that *together* form a product delivered to other participants in the Virtual Organisation - as noted in section I.4, we thereby do not care whether the resources all *belong* to the service provider or are actually aggregated through means similar to that of a VO[20]. Since performance of such service types can be considered critical with regards to the overall business goal(s), Application Services are generally subject to Service Level Agreements specifying the quality parameters, as well as the payment terms. To take countermeasures against failure timely, Application Services are often also constantly monitored and evaluated (supervised) - with complex aggregations this implies that all individual resources are monitored and the respective performance values need to be derived from this data.

It may be worth mentioning here, that a service provider can make use of the SLA support by TrustCoM to supervise his/her own resources, thus supporting preventive and automatic reconfiguration (see e.g. [19]).

It is furthermore up to the service owner to decide whether he/she wants to make use of the business process support of TrustCoM which will allow not only aggregation of the resources through a specified workflow, but also automatic adaptation of the business process details according to VO requirements, as detailed in the role definitions of the Collaboration Description.

VO Management Service(s)

The VO Management related service(s) take a particular role in the overall organisation and enactment of Virtual Organisations by managing and maintaining the participants, supervising the main processes and steering the lifecycle phases. Its main goal is to represent the customer's interest with the additional enhancement and capabilities to realise them. As such, the VO Management service differs from other participants in the Virtual Organisation, since it needs to ensure that all members can interact with each other and observe the overall and specific requirements, including policies, access rights and QoS definitions. Notably, this does not imply that the VO Management service itself needs to be capable of "understanding" all the related information, e.g. evaluation of SLA status information or interpret policy requests, but that the respective mechanisms are catered for in the VO and that the consequences are enforced accordingly.

These management functionalities may be realised through one or multiple services, either directly by the customer or through some intermediary providing the means to host VO Management services, i.e. acting as a "normal" service provider for these particular types of functionalities.

[20] Note that this is of no implication for the framework, though it may have legal impact, as e.g. discussed in section I.6

In its position, VO Management is the first member of a Virtual Organisation and the one service responsible for identifying the required participants for reaching the business goals as defined by the customer (VO Initiator). Likewise, it needs to provide the means for turning the goal description into a collaboration description that includes details about the roles that need to be manned for enactment. We do not presume, however, that such a service can realise this process which requires detailed knowledge about business processes, but that it uses supporting services for this task – rather, VO Management will take this elaborated description and extract from it the individual role definitions, including such requirements as QoS parameters and actual task descriptions. The service will furthermore take consequences for adapting requirements, respectively collaboration description, depending on the identified and available participants, i.e. whether all roles can be manned and whether the overall requirements can be fulfilled by it.

With this position, the VO Management service will not only have the responsibility to set up the VO and all involved participants, but acts also as the “contractual endpoint” of all SLAs (cf. section I.6), even if a participant is generally required to deliver a product / a set of data to another member rather than to VO Management.

Even though the VO enacts a business process in order to reach its respective goal(s), VO Management functionalities itself are not subject to a workflow engine (though it may be supplemented by it). VO behaviour is mostly defined through the means of policies that specify which conditions lead to reorganisation of the VO etc.

The full description of the VO as realised by VO Management is maintained in the so-called General VO Agreement (GVOA) that relates legal and electronic contracts – see section I.6.

IV.1.b General Considerations

For reasons of simplification and without loss of generality, we will furthermore assume that no service provider already supports any of the security, trust, or contract management aspects as provided by TrustCoM. This does not imply that such components may not be provided by the service owner him-/herself, but rather that the infrastructure may need individual configuration, if the component can not be integrated in the same way as the ones provided by TrustCoM - due to the strong SOA approach of the project, however, most pre-existing solutions may be plugged into the provided middleware without too much effort.

The setups discussed below reflect only TrustCoM’s *recommendations* and not real requirements. Accordingly, e.g. any Application Service Provider that wants to ignore authentication issues by *not* deploying the Security Token Service is principally free to do so, as long as this does not violate the overall conditions of the Virtual Organisation the resources are participating in.

When deploying the system, one always has to consider that non-regarding the flexibility of TrustCoM’s middleware, a maximum distribution of components is not recommended performance wise: the more often a functionality is required, like e.g. regarding messaging, the “closer” it should be deployed to interacting components.

We shall furthermore implicitly assume that a service provider does not object to the TrustCoM requirements and concepts, as otherwise he/she will not want to participate in a TrustCoM VO and thus not consider deploying the systems in the first instance.

IV.1.c Basic Setup

Regarding deployment of individual components, we may distinguish the following setup structures that roughly represent typical building blocks within the infrastructures of the various Service Provider types participating in a Virtual Organisation:

(a) the “gateway” structure: the interface between a TrustCoM VO and the individual Service Provider’s infrastructure, respectively his/her resources.

(b) the actual infrastructure of a Service Provider: hosting the actual services and/or resources of a Service Provider. Notably, employees are considered part of such an infrastructure.

(c) VO Management main structure: basic setup of Service Providers wanting to host VO Management capabilities

Within the following we will discuss the typical component deployment across a Service Providers intranet for the above listed building blocks.

1) The Gateway

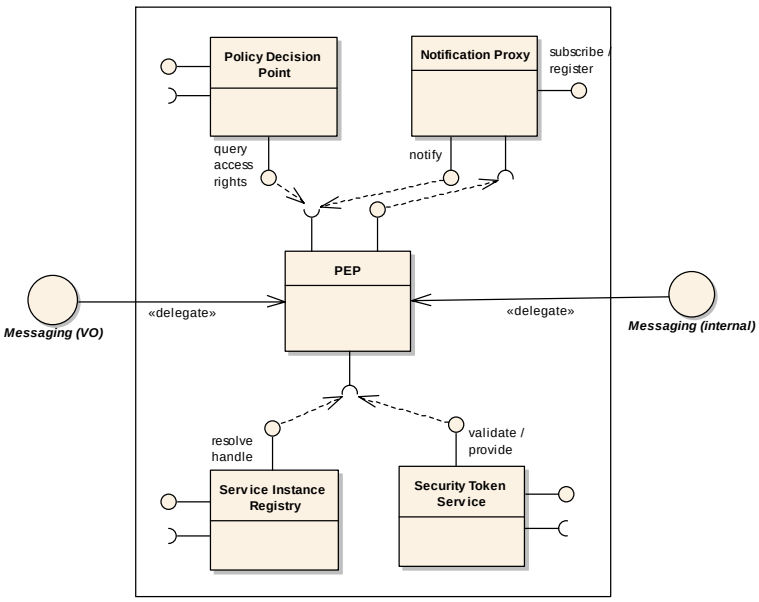


Figure 22: The Gateway and its components

The Gateway structure acts as the (virtual) endpoint of the Service Provider, i.e. its resources and/or products. Its main functionalities consist in enacting messaging related functionalities on the interactions between Service Providers in the VO. It typically incorporates the following components (cf. section III.2.b):

- the Policy Enforcement Point (*PEP*) intercepts messages and enacts upon them a message handler chain that may alter and/or block the message.
- the Policy Decision Point (*PDP*) verifies whether the message meets specific policies or should be blocked – this acts mostly as an access rights verification.
- the Security Token Service (*STS*) issues or validates security tokens, enabling the PEP to authenticate to other PEPs. To sign and encrypt messages, as well as validate and decrypt them upon receipt.

- the Service Instance Registry (*SIR*) redirects messages according to the handle provided in the message header
- the Notification Proxy sends and receives notifications

Deployment Considerations

Almost all participants in a Virtual Organisation need to host a structure providing functionalities similar to the gateway, so as to realise TrustCoM compliant messaging – in particular with respect to authentication.

Being the main contact point of the SP for both incoming and outgoing messages, the Policy Enforcement Point is principally the only component that requires internet connection and infrastructure accessibility. However, in order to enact the specified functionalities upon the message, it is required that the PEP has direct connection to the according components, i.e. the PDP, STS etc. respectively the other way round (e.g. for the Notification Proxy).

Accordingly the functional components could be hosted by any machine as long as a connection between the component and the PEP is granted. As such, following the SOA approach, these components could principally be outsourced to third parties – however, this is only of limited practical use as shall be discussed here:

(a) duplicating security issues: as messaging between services is secured by means of the gateway structure (STS), communication between the PEP and an outsourced component would either require a different means of security or yet another intermediary gateway structure.

(b) delaying interactions: since communication between the PEP and the functional components is basing on SOAP like any Web Service interaction, any bandwidth limitations will seriously delay the overall processing of the gateway which implicitly delays all messages between participants in the VO.

(c) confidentiality issues: main task of the gateway being to enable security, confidentiality and privacy issues with respect to the SP's infrastructure, any outsourcing of such information is implicitly entrusting third parties with confidential information about the infrastructure that would otherwise be hidden in TrustCoM.

Generally, it is recommended to keep gateway specific functionalities within the SP's domain, though this does not necessarily imply that the domain may not be distributed across the internet (like e.g. with international companies) where proprietary security means (like VPN) are enacted between the according sub-domains anyway, thus overriding the security and confidentiality issues listed above. However, administrators aiming for such distributions should carefully evaluate the impact of the implicit delays.

2) SP Infrastructure

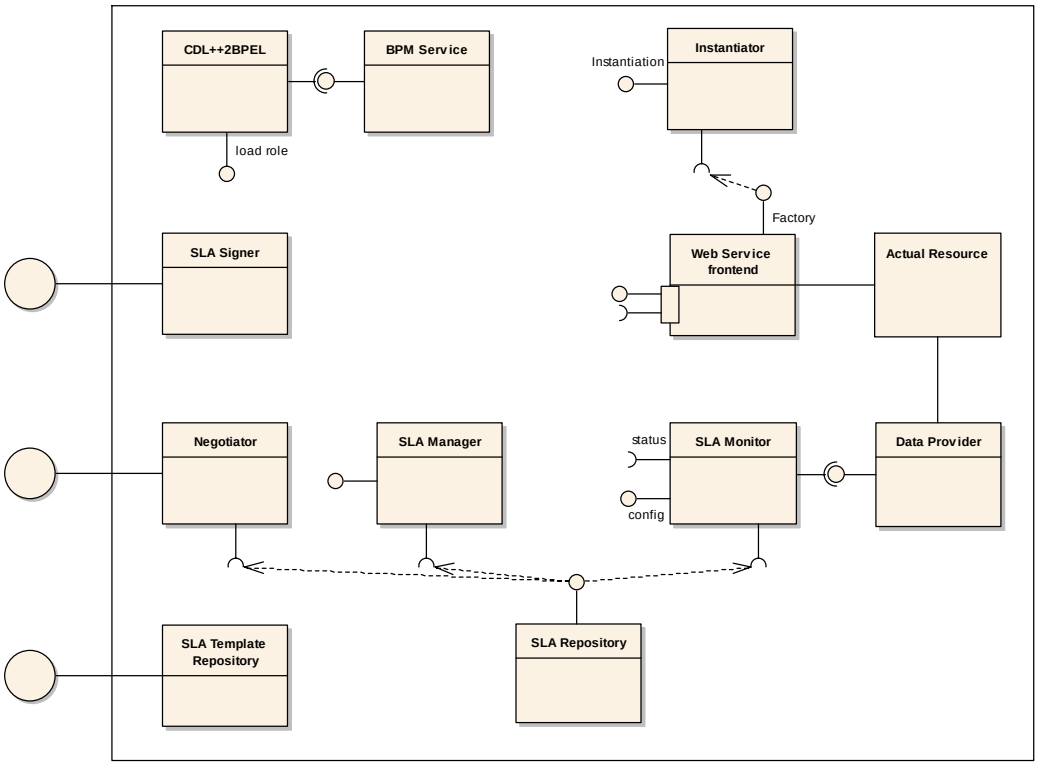


Figure 23: The fully configures SP infrastructure

The actual infrastructure of the Service Provider is “hidden” behind the gateway which acts as the main interface between this infrastructure and the actual VO. Accordingly, none of the resources as maintained and offered by the SP need to be exposed directly to the internet and connection needs only exist indirectly through PEP.

Next to the resources provided by the SP, the TrustCoM framework foresees a number of infrastructure specific components that support the Service Providers tasks:

- The *CDL++2BPEL* component takes the role descriptions and adapts local workflows according to the role specific requirements so that they can be enacted in the *Business Process Management Service*
- The *SLA Management* related functionalities ensure that SLAs can be published, negotiated and enacted, including the monitoring of the performance
- In order to coordinate instantiation and configuration of the SP’s resources, an *Instantiator* exposes a simplified management interface

Deployment Considerations

Independently of the actual resources a Service Provider will offer to the Virtual Organisation and the support he/she chooses to integrate from the TrustCoM framework, the distribution of components across the local infrastructure will have to observe specific issues to allow optimal functional support:

(a) connectivity to the gateway: almost all components deployed on the local infrastructure will have to be accessible (and have access to) the gateway structure,

if they intend to interact with other participants in the VO. Note that they do not need to *know* the location of the gateway though.

(b) exposed components: as an exception to the prior statement, there are some components which can not rely on the gateway specific functionalities, simply due to the fact that the gateway will not be configured properly at the time their functionality is required – an example of such components would be the SLA Negotiator, which is accessed prior to setting up the Service Provider. Obviously, such components could be easily outsourced to Third Parties, insofar as they do not require immediate access to SP specific and potentially confidential resources.

(c) gateway-independent components: a number of components used by the Service Provider do not require access to the Virtual Organisation and/or are only used by other resources *inside* the infrastructure. This holds true in particular for those resources that realise intermediary steps in the business process that is enacted to deliver the SP's "product", i.e. that have no functionalities exposed to the VO.

(d) messaging delays: like with the gateway specific components, it has to be taken into consideration that any bandwidth restrictions between components / resources may lead to serious delays in overall progressing, increasingly so the more often the respective functionality is required. As such, components that need to interact often should be placed inside the same subnetwork.

(e) virtual endpoints vs. real endpoints: with the gateway's specific functionality of "virtualising", resources seemingly exposed to the VO, do not necessarily have to be in direct connection to the PEP, as they may be encapsulated by a BP engine or through a Web Service frontend etc. that redirects the messages to the actual resource(s).

Due to the specific setup, principally any component that interacts through a gateway can be outsourced to a third party, given that they host a similar gateway infrastructure – the gateway specific functionalities will ensure that message redirection and security enactment will take place according to the Service Provider's requirements. However, as soon as the component requires interactions with other infrastructure specific resources, the impact on message delivery speed should be estimated carefully.

As opposed to this, non-gateway dependent components, such as the SLA Negotiator, which *may* be outsourced to third parties without an intermediary gateway structure – however, it has to be carefully considered what kind of information and interaction the according functionality requires. As such it is not recommendable to outsource e.g. the negotiation capabilities as generally human interaction will be required to validate the offers and counter-offers. On the other hand, the SLA Template Repository may be easily shifted to a Supporting Party as it hosts data that only requires irregular updating.

Furthermore, there may be a number of components that are infrastructure specific like in particular the Data Provider. As discussed in Appendix A to this document, this is not a TrustCoM particular component and *needs* to be hosted at the same site as the resource that is to be monitored.

3) VO Management Main Structure

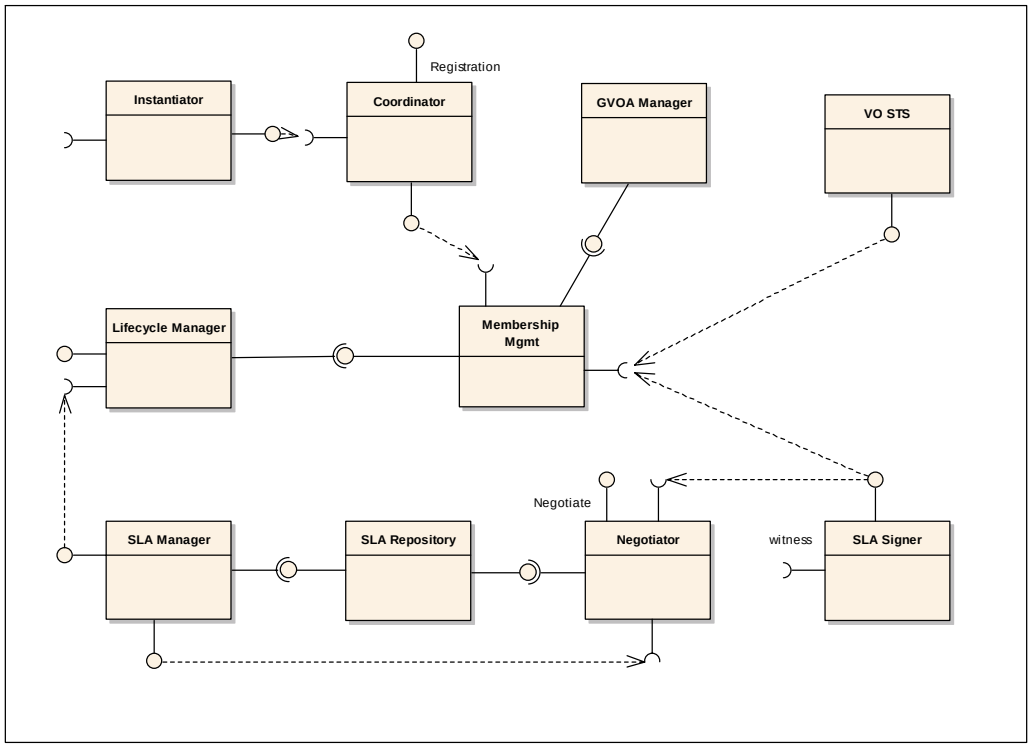


Figure 24: The main components of the VO Management Structure

In principal, the VO Management service hosts the same structure like any Application Service or Trusted Third Party, i.e. the main resources on the local infrastructure, hidden behind the functionalities of the gateway. However, the VO Management host does not provide business logic resources of its own, but basically only TrustCoM related functionalities that are required for steering a Virtual Organisation.

This basically covers functionalities with respect to

- managing the members inside the VO (*Membership Management*)
- triggering the actions with respect to the individual lifecycle phases (*Lifecycle Management*) and distributing configuration information in a coordinated manner (*Instantiator, Coordinator*)
- maintaining the overall contract (*GVOA Manager*)
- negotiating and configuring the SLA related functionalities (*SLA Management*)

Deployment Considerations

Generally, the VO Management Infrastructure faces the same issues as the SP Infrastructure – accordingly the same considerations apply here as in the previous section.

Note that the components of the VO Management Infrastructure take primarily management functionalities and as such need to be secured accordingly. Most of these components are *principally* independent from each other in the sense that they do not need to be hosted in the same domain, as long as the respective Service Provider's gateways are configured to allow each other access. Likewise, VO Management may be distributed

across the VO according to their functionalities or even by splitting up the information content between multiple instances of the same component.

Like with any other structure, any distribution of components / information has to take potential messaging delays into consideration.

IV.1.d Deployment Recommendations Overview

The following table summarises the above discussions by assessing for each component involved in the TrustCoM framework where and how it may be deployed. As was discussed in this chapter, most components may be used in various forms on different locations, e.g. the SLA Monitor may be used both as part of the Application Service, as well as a Trusted Third Party itself, whilst it may be deployed on any location within the intranet, as long as the issues detailed above are maintained (communication with the internet etc.).

Within the table, we identify which components are recommended to be deployed by the respective Service Provider Type ("type), like e.g. that any provider within the VO (TTP, ASP and VOM) should deploy the gateway-specific components, whilst a Supporting Service provider does not require these functionalities.

In addition to this information, the table lists *where* the respective components may be deployed so that they may be used by the provider ("location"), whereas we assume that the Infrastructure has a connection to the Server and the Server itself to the internet (and thus the VO). We furthermore list "standalone" to distinguish functionalities that can (or should) be offered as independent services (Trusted Third Parties or Supporting Services).

In the table,

> O stands for optional: deployment is possible but not required, so it is up to the service provider's discretion
>
> R for recommended: i.e. not necessarily deployed, but advisable with respect to TrustCoM' functionalities
>
> N for necessary: needs to be hosted at least in a similar way – these components address vital functionalities for the full TrustCoM capabilities, such as security.

Implicitly, no entry means that the service can not be deployed in this location, respectively for this type of service provider. Note that the distinction between Server and Infrastructure location is weak, so that in most cases what is recommended in the one location is optional for the other (and falls together whenever the Service Provider only hosts one machine, i.e. where Server and Infrastructure are identical).

The setup should be considered a "recommendation" rather than a constraint, as configurations are thinkable, where e.g. the SLA Monitor is hosted by VO Management or where the BPM service is deployed on an additional machine in the infrastructure. The rationale for the recommendations listed below can be found in the preceding sections. The components are only to be considered a "requirement" for deployment if the service provider wants to make use of the respective functionality without providing his/her own components to substitute the TrustCoM framework.

Component / Service		Provider Type				Location				
		Supporting Service	Trusted Third Party	Application Service	VO Management	Server (Gateway)	Infrastructure	other Machines	standalone Supporting Service	Standalone TTP
VO Mgmt	VO Lifecycle Manager[21]			O	R	O	R			
	Membership Management			O	R	O	R			
	GVOA Management				N	O	R			
BP Mgmt	BP Repository		O	R		O	R	O		O
	CDL++2BPEL		O	R		O	R			
	BPM Service		O	R		O	R	O		O
Trust & Security	Reputation Management Service								N	
	Reputation Evaluator / Scoring System									N
	Security Token Service		R	N	N	R	O			O
	Secure Audit Log	O	O	O	O	O				N
SLA Mgmt	SLA Template Repository		O	O		O			R	
	SLA Negotiator		O	N	N	N				
	SLA Manager		O	N	N	N				
	SLA Signer		O	N	N	N				
	SLA Repository		O	N	N	O	R	O		O
	Notary									N
Policy	Policy Service									N
	Policy Decision Point		R	N	N	R	O			O
Infrastructure Support	Service Repository (UDDI like)								N	
	Discovery Service				O	O			R	O
	Policy Enforcement Point		N	N	N	N				
	Service Instance Registry		N	N	N	R	O			O
	Notification Proxy		N	N	N	R	O			O
	Notification Broker									N
	Instantiator		N	N	N	O	R			

table 1: overview over components and their deployment
(O: optional, R: recommended, N: necessary)[22]

[21] The combination of the VO management tools is often referred to as the VO toolkit in this document.

[22] Note that this table does not reflect which *functionalities* (and hence components) are required for realising the TrustCoM capabilities, but which *deployments* are possible – as such, even though e.g. VO Lifecycle Management components / capabilities are *necessary* for the TrustCoM framework, deploying them at a separate VO Management location is not *necessary*, but only *recommended* for the sake of message

IV.2Business Model Specific Deployments

The business models as discussed in section I.2 of this document are realised through different general deployment structures of the underlying TrustCoM framework. This section will discuss the basic deployment issues with respect to these business models, thus giving an indication of the flexibility of the TrustCoM framework.

Note that though the framework principally supports all the business models, not all of the components distributions are considered equally sensible due to implications on issues like security, confidentiality and execution speed. The respective considerations are highlighted where not discussed in previous sections.

IV.2.a One-to-One and One-to-Many Models

Model 1B of the business models foresees a hierarchical VO structure where each participant forms contracts and interactions with any number of companies according to their respective needs, whilst only the "highest level" participants enter a contract with the VO representative (VO Management) – this is identical to the subcontracting model discussed (amongst others) in section I.4. Whilst it is up to the VO Initiator to decide to which level of detail to define the collaboration and hence the hierarchy of the VO, only the "highest level" participants are directly responsible for their performance in the Virtual Organisation, whilst all other participants act as subcontractors that hence report to these "highest level" entities. Technically speaking, subcontractors are only indirectly participants of the TrustCoM VO and thus have to make use of the TrustCoM framework only in a limited way.

Implicitly, the contractor will generally invite the subcontractors him/herself, i.e. the collaborations are not fully detailed by the VO Manager, respectively VO Initiator. Since the VO Manager does not enter direct contracts with the subcontractors, the contractor takes over responsibility for the respective performances. The relationship between contractors and subcontractors may thus be regarded as a relationship similar to the one between VO Manager and main contributors ("highest level" participants) – as such, these collaborations may be realised using the TrustCoM VO structure, i.e. as VOs on their own.

With the latter approach, each Service Provider in the collaboration would deploy the gateway and infrastructure according to his/her respective requirements and additionally, each SP that acts as a contractor would also deploy VO Management support. Note that the strict business model does not foresee any outsourcing of components to Trusted Third Parties.

Model 2B depicts the "standard" TrustCoM VO, similar to the model that has been used for deployment discussion in all previous sections as it describes the default structure from which all other models can be derived. According to this model, the VO itself forms a flat hierarchy with all VO participants, whereas the VO initiator is represented through an intermediary entity that takes all responsibility for the members and their performance. This

---

exchange etc. (see text). Please refer to chapters II & III for a discussion on the relevance of the individual components.

VO Manager host enters contracts *per partner* on the one hand and with the VO Initiator on the other hand, so that it seemingly forms a single entity that aggregates the whole VO processing.

This approach foresees Trusted Third Party support that may take over any set of tasks that are considered relevant as *TTP* services by VO Management (cf. discussions in sections 0 and IV.1). Main deployment issues of this approach are:

VO Management is hosted by a neutral third party and not by the VO Initiator or a Service Provider (see section on Partner Managed Consortia below).

Main data files and infrastructure support (brokering, logs etc.) are hosted by TTPs and are thus outsourced from the individual participants.

The participants are configured as plain Service Providers (gateway and infrastructure as discussed above) – potentially with outsourced infrastructure support, such as the Notification Proxy. Note that a TTP hosted gateway is thinkable, though not advisable for confidentiality and security reasons.

The SLA Evaluator is hosted by a TTP as suggested by the SLA Management subsystem architecture (Appendix B, chapter III).

Both **Model 1B** and **Model 2B** assume that all participants in the VO come from the same (Enterprise Network) background, i.e. with common basic consortium agreements. Note that this basic consortium agreement could exist prior to the VO (EN agreements) or otherwise be part of the common contract details in the GVOA. It is of no direct importance to the TrustCoM framework whether the consortia brings in its own means of interactions, message security etc. as the gateway may be configured so as to use different communication modes and/or skipped entirely if so desired. For realising the TrustCoM VO specific functionalities and thus for dynamic *management* of the VO, the base deployment as discussed in the preceding sections is required.

IV.2.b Trusted Third Party Consortia Models

With **Model 3B** and **Model 4B** we depict the case where the VO members come from multiple different consortia and thus do not necessarily share a common agreement when entering the VO and potentially even during the operation of the collaboration. This is identical to maintaining different clauses per consortium in the GVOA. Note that it is possible for each consortium to use a different set of TTP services for supporting tasks so that a different trust bases may be exploited, even though this is not explicitly stated in the business model.

Model 3B in specifically foresees that Trusted Third Parties generally take over management of the individual consortia and their according messaging requirements as defined by their respective consortia. The TrustCoM framework can address these issues in multiple ways:

TTPs may take act as additional gateway structures that each encapsulate a consortium thus indirectly forming a hierarchy of resource providers, even though each service provider is still responsible for delivering his/her products on his own, i.e. contracts are formed between VO Management and each participant, not between gateway provider and each provider – implicitly the gateway provider does not take over responsibility for the SPs' performances, but only for enacting the relevant messaging specifications, security

requirements, interaction redirections etc. The basic setup of each participant is left unchanged, whilst the additional gateway provider(s) only deploy the gateway structure, using the local Service Instance Registry for correct redirection within the consortium.

If no additional messaging and confidentiality requirements need to be addressed, separate consortia are maintained by the combination of membership management and GVOA support. VO Management is not restricted to a single consortium and according setup information, but may maintain and enact multiple configurations at the same time.

Extending this model with multiple end-users (**Model 4B**) implicitly defines a many-to-many model where a set of customers exploit the capabilities of the Virtual Organisation (which may consist of multiple consortia). Note that a Virtual Organisation is generally encapsulated as a single entity by the VO Manager so that the direct approach would consist in one (or multiple) VO Management services for providing the VO's capabilities to the customers – implicitly, this forms only indirectly a many-to-many model.

With multiple VO Management instances, the simplest approach consist in deploying multiple VO Management instances that all make use of the same information sets hosted at a third party. This basically represents a sharing of functionality with only one main data set (which may be replicated for security reasons) – this setup is discussed in section 3 of chapter IV.1.c.

Depending on the business model details, each customer may principally refine his/her own requirements with respect to VO enactment, so that the collaboration adapts to the according needs. A similar issue is addressed by the AS testbed (cf. section V.2) where a general VO agreement may already exist between all participants before the actual details *per customer* are defined. In the simplest case, customer requirements are restricted by and may be derived from the general VO agreements so that no additional negotiations (and hence SLAs) are required for fulfilling the respective needs. Implicitly, no new VOs are formed but just "subconfigurations" distributed, such as interaction specific requirements, workflow details etc. There are no additional deployment issues to be addressed besides for the fact that the VO Management host will have to provide a portal like interface to the user to allow access and interactions.

IV.2.c Partner Managed Consortia

Finally, **Model 5B** and **6B** depict the case where participant or customer consortia host (and hence cater for) VO Management capabilities. Whilst this has an obvious impact on the legal side related to the GVOA issues discussed in the previous sections, it does not have a direct influence on the underlying deployment model: since Trusted Third Party services, as well as VO Management capabilities may be derived from any kind of Enterprise Network with the according set of EN agreements. From the perspective of the two models, the underlying network is either part of the consortia the VO Initiator / customers (**5B**) or the VO Members (**6B**) participate in. Note that there is no technical obstacle towards customers and participants coming from the same consortium.

V Examples of Virtual Organisations

This chapter will present the TrustCoM framework as discussed in the preceding sections in the context of the two testbeds, namely the collaborative engineering and the aggregated services (elearning) scenario. These scenarios will exemplify the actual application of the framework in a more high-level overview, i.e. without re-discussing the architectural details – please refer to chapters II-IV for such information.

V.1 TrustCoM in the Context of Collaborative Engineering

The basic application scenario is as follows. A designer within an engineering consultancy wishes to improve the performance of a design- say the outer surfaces of a car body. He wishes to minimize the drag subject to certain constraints- eg, passenger size constraints, materials etc. He understands that this type of optimization problem can be performed using services hosted on an Enterprise Network. He is also working to time constraints, and requires service providers who have a certain level of performance.

He therefore specifies the composition of the VO that he requires- the collaborative business process, the levels of service he requires, the policies for dealing with exceptional events and so on. The TrustCoM framework then goes into action and candidate services that can fulfil roles within the collaborative business process are discovered and SLAs are negotiated with them. This discovery takes into account not only the functional requirements as defined by the choreography, but also factors such as Reputation to eliminate poorly performing service providers. The members of the VO are bound by the GVOA that is set up for them, it is signed and the VO is initiated.

Figure 25 shows the overall choreography for this optimization process. The overall process is essentially cyclic, with new geometries being evaluated if the target design performance is not met. If the design goal is achieved, then the overall process ends.

A point to note is that each organization that fulfils that required role may require internal ‘private’ processes to support it’s external behaviour for that role. A good example of this is data translation and filtering that may be enacted on the ‘Geom role’, whereby data is changed into another format before it can be processed by the application specific components within that service.

The SLA requirements are that the simulation service must ensure that a certain proportion of CPU capacity be reserved for the client. The SLA for the PDD must also provide limits on the acceptable access times for retrieving data. Similar SLA requirements prevail for other services as well. If the SLA is violated, then the organisation is removed, its reputation is decremented, and a replacement service provider that agrees to the GVOA is enrolled.

The VO is also regulated by policies- such as a policy that specifies that if the reputation of certain critical services (eg, the Flow simulation service) drops then the service becomes more closely monitored. For example, the tempo of SLA monitoring (in this case, for the level of CPU used) may be increased and more frequent evaluations may be made.

Once the design optimisation is complete, the client is notified and the collaboration ends. The VO goes into termination mode, remaining duties (such as payment and re-

configuration of security systems etc) on member organisations are enacted and the VO agreement is terminated.

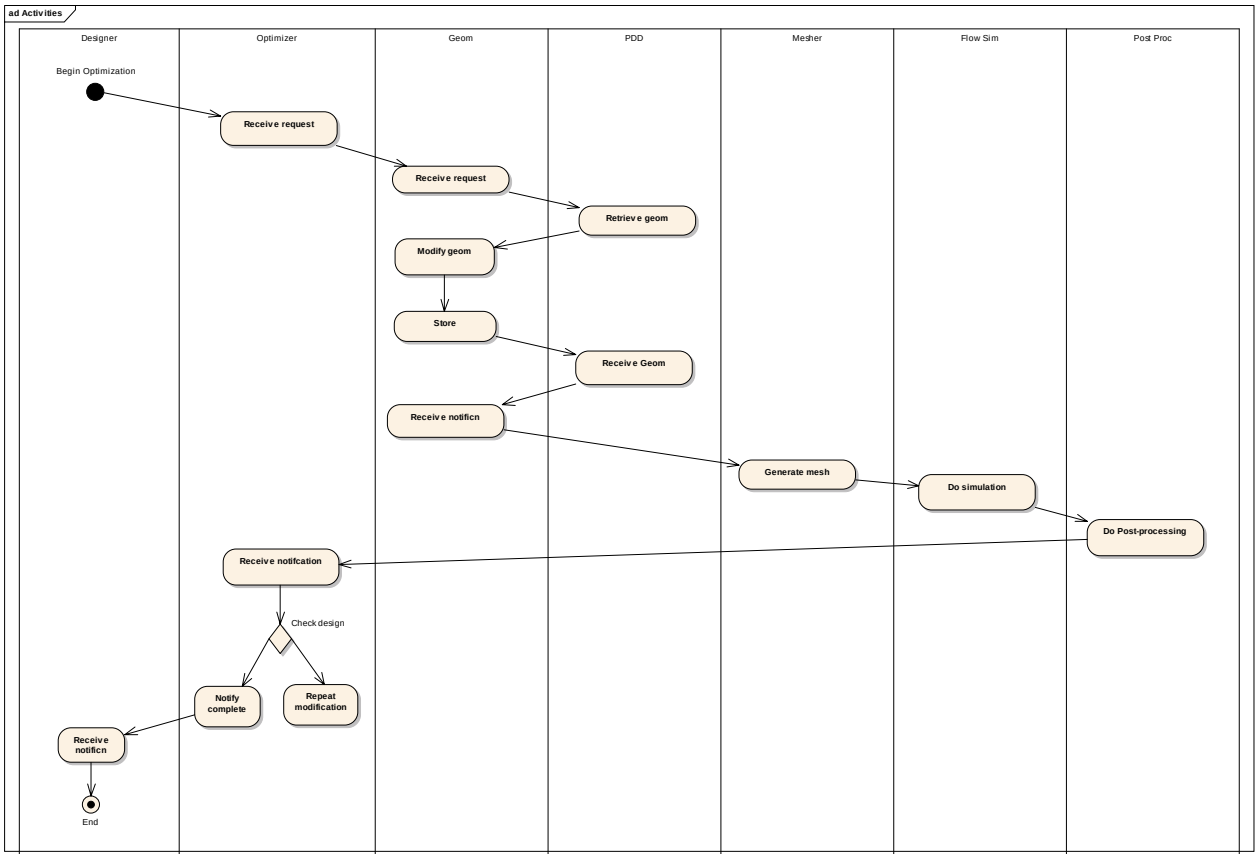


Figure 25 Scenario Choreography

V.1.a Deployment Overview

To simplify matters, we first use the same application scenario as described in the previous edition of this report [D62]. The authors are satisfied that this represents a generic use case in Collaborative Engineering using the kinds of application services that are likely to be used in the near term. Variations on this use case reflecting different business models is presented below.

The basic deployment views- service and component based- are basically unchanged from those presented in the previous edition of the report and are not affected by the work done to date in AL2. Therefore, they will not be presented here. Instead, we present some lessons learned from the work done in AL2 and discuss the particular IBM Business Models that fit this particular scenario. This discussion is concentrates on how the software deployment may be affected by the business model that is adopted.

Deliverable D54 summarises the experiments that were performed in the CE Test Bed. These experiments consisted of:

1. Service and message security using the Messaging Infrastructure, STS and PDP Security components

2. Reconfiguration of a VO in response to changing service provider performance- ranging from increased monitoring to member replacement (from the point of view of SLA)
3. An experiment in Business Process Enactment.

These experiments in AL2 have provided the authors with further understanding of the implications from deploying Trusted Third party components. In particular, we see that *configuration* as well as *deployment* becomes the primary issue.

The following configuration issues became apparent:

1. A TTP may impose significantly different security requirements from those of the VO members. An example of this is the security requirement for the SAWS service, which requires certificate-based SSL rather than token-based message-based security model that is used by the VO Partners.
2. The Policy Service is an instrument of the VO Management sub-system and needs to abide by strict security requirements.
3. The Notification Proxy also needs to ensure that VO event information is protected in transit from unauthorised users, eg, within organisations external to the VO.

In Issue 1, the deployment implication is that some prior SSL configuration is necessary. In this case, a PEP belonging to the Gateway of a particular partner would have to be configured to interact with the SAWS.

Issue 2 refers to the Policy Service reconfiguring a service provider's PEP. In this case, the Service Providers would have security configuration that would allow or deny access to the Policy Service. Also, the Policy Service may be entrusted to store sensitive configuration information in its policy store, for example, PEP handler configuration policies.

In Issue 3, it is also possible that certain partners may prefer the event information to be protected from other VO Members due to its sensitivity. This may lead to access control requirements on the notification topics and the appropriate configurations.

There are a number of deployment options for the TTPs:

1. They continue to use their own preferred security infrastructure but with prior configuration (eg, SSL configuration as in the case of the SAWS)
2. They use the same TrustCoM security sub-system components as the VO- eg, the STS, PDP but with their own policies.

In summary, it appears that Virtual Organisations would have to be prepared for a significant amount of configuration and initialisation steps depending on the security requirements of the TTPs they decide to work with within the VO.

V.1.b Business Models

We now consider the effect of alternative business models on the software deployment for this VO.

The business model adopted by the scenario can be summarised in the following diagram:

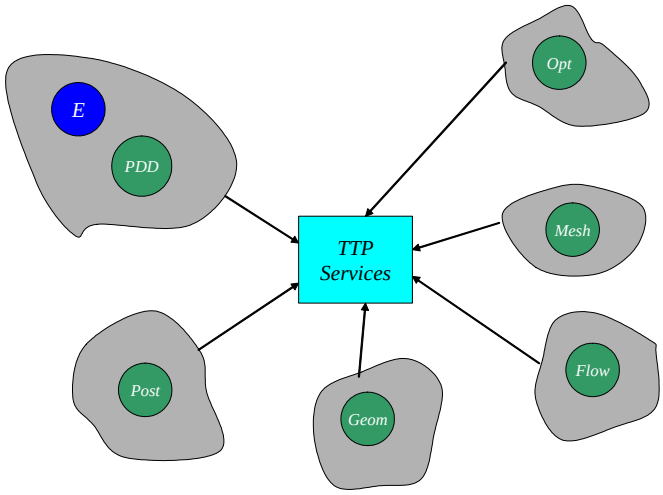


Figure 26 Current Business Model adopted in the CE Scenario

Note that each grey area denotes a boundary of an organisational entity. This denotes both a 'legal boundary' and a 'intranet boundary' for the partner within the VO. In this model, each partner company prefers to focus on a particular core capability. Interactions are mediated using the messaging sub-system hosted by the EN/VO and value-adding services are provided by the Trusted Third Parties.

It should be noted that if this particular business model were to be adopted then various contract clauses would have to be introduced to limit liabilities in the case of partner's defaulting on their required interactions. The reason for this is due to the nature of the collaborative business model (Section V.1, Figure 38 in [D62]) whereby a partner is critically reliant on other partners in order to deliver its own expected contribution to the VO.

This particular problem may be resolved by either the contract solution suggested above or by forming a limited Joint Venture for the purposes of the collaboration. The JV would be a single shared legal entity that eliminates this liability by sharing the risks and the profits from fusing the two capabilities together. Note that this does not necessarily imply full merger or acquisition of one company with another.

It is clear that variations on this model are possible for different consortia models:

1. some of the partners may collaborate to form JVs to eliminate the aforementioned liability issue (model 3B), and/or
2. the Trusted Third party services may be hosted by a consortium of either Enterprise customers or a consortium of suppliers (IBM Business models 5B or 6B).

The main criteria for forming this consortia/JV in Option 1 would be:

1. realising a new business capability that cannot be achieved by traditional client/supplier models,
2. exploiting new or existing markets that cannot be addressed by a single company alone

3. reduction of costs through aggregation of shared infrastructure, services and skills

In the case of option 2, it is possible that a new set of TTP services could be provided that address the needs of a particular market. In the case of BAE SYSTEMS, TTP services may be hosted within the EXOSTAR electronic market place to facilitate collaborative engineering projects. Other sectors (eg, automotive, air carrier, pharmaceuticals etc) could extend their existing market places in a similar way.

Examples of TTP that could be hosted in this business model include:

1. VO Management services
2. Policy Services, and
3. Security Token services

V.2 TrustCoM in the Context of eLearning

The Aggregated Services testbed bases on an eLearning scenario that makes use of enhanced features of Metacampus. In principle we presume the following situation: a student wants to make use of the internet for learning courses according to his/her[23] individual needs, covering not only the learning goal, but also the current knowledge of the user with respect to the topic. He is expecting a lesson with individual learning support and is aware of his restricted learning time – accordingly he wants fast on-demand performance and high quality training courses.

In order to receive such a learning course, he contacts a “VO Learning Portal Service” that provides the capability of arranging a set of learning providers from a “Learning Enterprise Network” so as to provide the bespoke lessons. The VO Learning Portal Service acts as a user front-end to the VO Management capabilities and as such takes over the responsibility for setting up a collaboration of learning resource providers tailored to the student’s respective needs.

The student will provide information about his learning goals so that the VO Learning Portal can query the right Training Consultant service to identify the necessary requirements from both the student’s side, as well as from the learning providers’ side. The user’s requirements cover in particular issues related to his background knowledge regarding the lessons, i.e. where the lesson should start and what scope it should have. This information implicitly defines the learning course details and as such the “workflow” guiding the learning providers.

With this list, discovery is initiated to identify learning providers that provide the respective lessons, but also meet the requirements regarding SLA (incl. time restrictions, quality and budget) and potentially also reputation of the services. SLA details and actual availability may be negotiated before setting up the Virtual Organisation, depending on the specific requirement details. From the SLAs, the requirements list, a general VO Agreement (GVOA) is set up that legally binds the participants to the VO and its specific conditions & terms.

[23] Furthermore addressed as male for reasons of simplicity.

Figure 27 depicts a simple learning path workflow with 4 learning resource providers. The progress, i.e. the individual tasks in the workflow, basically depend upon the progress of the student and can be compared to individual learning lessons. The learning process is finished once the student passes the 4th learning provider of the workflow, or the process otherwise fails (due to lacking learning providers, network breakdown or similar).

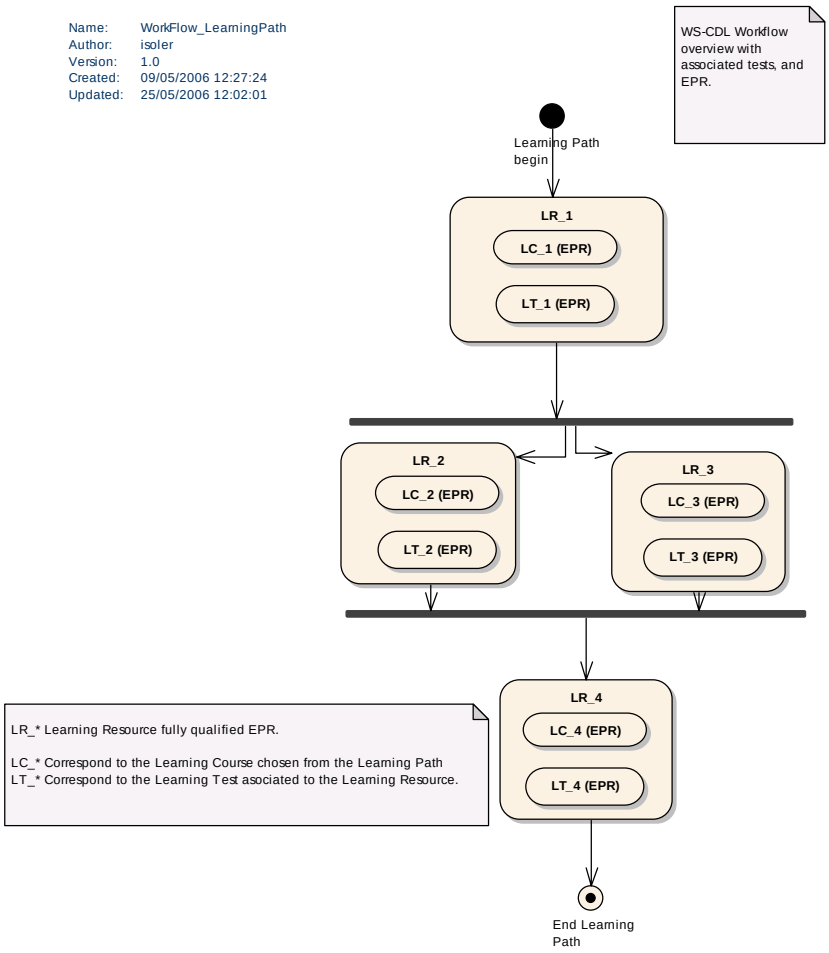


Figure 27: Workflow of a Learning Path

Since response-time of the individual learning providers to user requests is critical to optimally support the learning process of the student, the individual SLAs foresee strict rules on response-time, so that potential delays can be compensated by replacing the respective provider. Such quick replacement is possible assuming a uniform underlying Learning Enterprise Network in which multiple resources of the same type are available.

V.2.a Deployment Overview

Within this section we will detail the framework configuration for above application scenario in line with the architecture descriptions in chapters III and IV. We will assume here that the setup of the underlying EN/VO infrastructure is identical to the configurations detailed in chapter IV, i.e. the individual participants provide a gateway structure, instantiation and monitoring capabilities according to their respective service type (TTP, ASP or VO Management).

It is worth mentioning here, that the individual participants, unlike in the CE scenario, do not provide abstract, aggregated “products”, but rather the services (lessons) as they directly map to resources and functions in the respective infrastructure. Accordingly, the participants do not host their own business process engines and the overall learning path is steered by the VO Manager.

Within the scenario, we can distinguish the following services according to their type (cf. section II.1.a):

- VO Management Service(s):

The task VO Management is two-fold with respect to this scenario: on the one hand it indirectly acts as the interface between user and Virtual Organisation by facilitating the VO Portal (see Application Services), on the other hand it steer and manages these interactions on basis of the learning path and the student’s progress. Besides this, the VO Manager takes over the typical VO Management tasks, as detailed in chapters III and IV.

Since in this scenario, the main tasks relate to the functionalities of the Metacampus, the underlying business model is identical, thus rendering former Metacampus’ hosts to VO Management hosts.

- Application Services:

The main application services in an eLearning environment consist in the individual Learning Resources that provide the lessons and tasks to the student. These services act on a request-response-basis, i.e. without triggering other application services themselves.

In addition to the learning resource providers, one of the main application services in the eLearning testbed consists in the WS-Gateway that builds the actual interface between the VO Manager and the student. In our approach, the WS-Gateway makes use of a portal (“VO Learning Portal”) as a front-end to the user. The gateway enacts the VO’s security requirements and redirects messaging according to the VO Management instructions, so that the student always interacts with the current lesson provider.

- Trusted Third Parties:

As recommended in section IV, the VO hosts the most typical “neutral” services as TTPs, namely Discovery, Logs, SLA Evaluators, Reputation Evaluators, Notaries and the Policy Service.

In addition to these, the testbed foresees a set of eLearning specific Trusted Third Parties that support the application services tasks. These are in particular resource-like services that provide state information related to the student: since this data may be shared between different VOs (e.g. if the student takes more than one course at a time), these services could in principle be considered Enterprise Network (EN) wide Supporting Services. However, information in these repositories needs to be treated confidential and may require explicit user permission for access. Accordingly, either the EN needs to enforce security settings on its own, meaning that all EN members implicitly have the right to read the user information, or alternatively that reading permission is issued per VO.

Within the scenario, we presume that access rights are issued by VO Management. It is worth noting that these services take an intermediary position between Trusted Third Parties and Supporting Service, since it may not really require VO specific (re)configuration.

These services are namely (cf. Figure 28):

1. UDB Service: storing the user data base
2. Vocabulary: lists the values from the eLearning portal
3. TC_Vocabulary: maintains qualifications, languages, competences, jobs etc.

- Supporting Services:

Besides for the special position of the services just mentioned before, that may be considered both Trusted Third Parties and Supporting Services, the scenario makes use of the following Supporting Services:

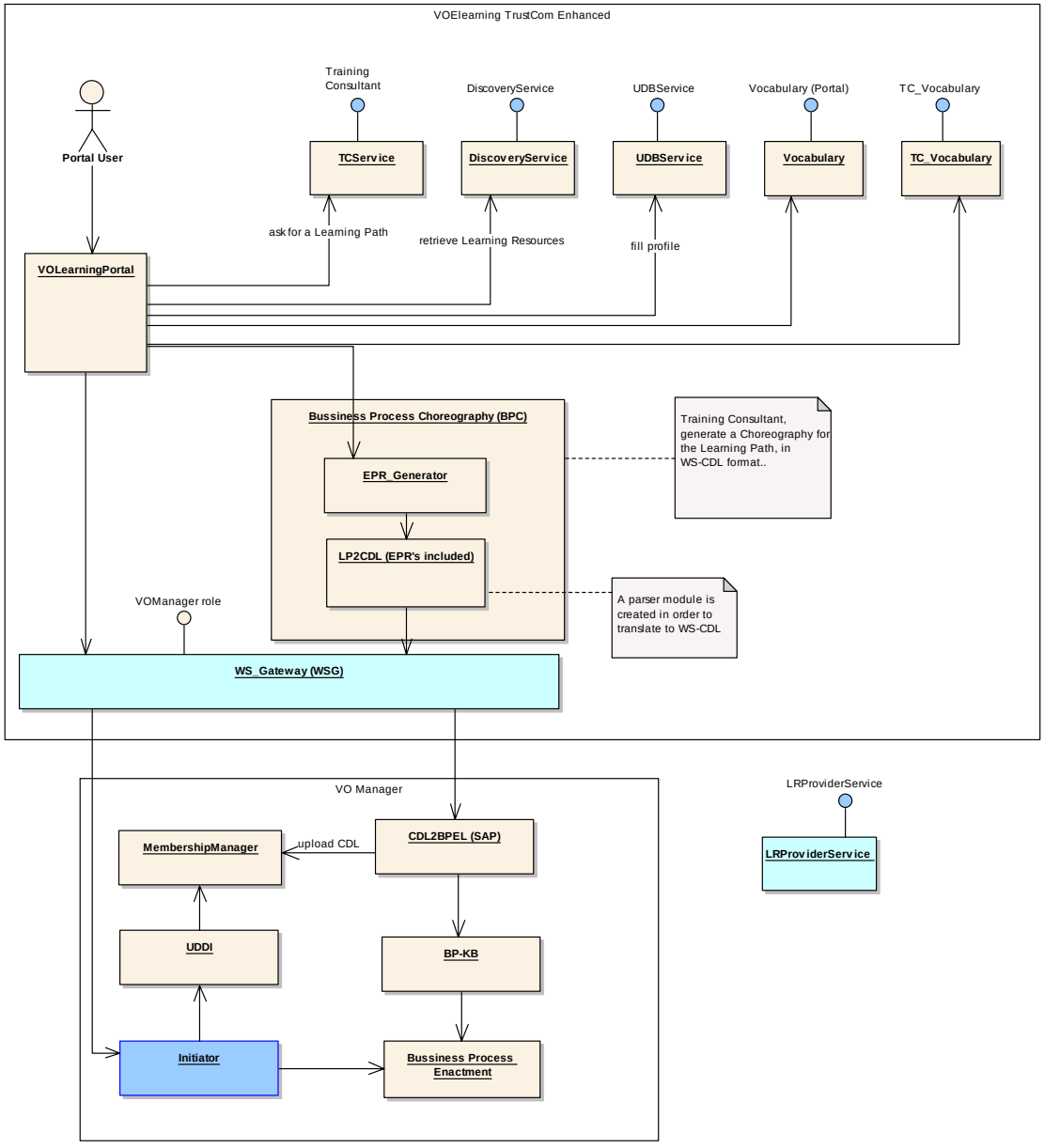


Figure 28: Overview over the services involved in the eLearning scenario

A Training Consultant (“TCService”) that is in principle identical to the Business Process repository as detailed in previous sections. The “collaboration description” of the Training Consultant is much clearer defined, however, as the influences from user requirements and the Service Providers capabilities are more restricted than in an engineering case.

Furthermore, the Enterprise Network is assumed to host the three service types related to discovery, namely the Service Registries, the SLA Template Repositories and the Reputation Management Service.

The picture above (Figure 28) provides an overview over the services and some of the components involved in the application scenario described in this section. Note that this diagram does not depict the full service structure with respect to deployed component, as this would simply reproduce the diagrams from chapter IV.

In the following sections we will discuss specific deployment issues with respect to individual subsystems in more detail:

- VO Management:

As noted, VO Management capabilities are shared between the actual VO Manager and the WS-Gateway. However, main task of the WS-Gateway consist in redirecting messages according to the current position in the training path. As such, the WS-Gateway is in principle the outsourced gateway of the VO Manager Service (cf. Deployment Discussion, chapter IV).

The WS-Gateway *must* be capable of enacting authorisation and identification and *may* be capable of issuing security tokens and access right policies itself to adjust authorisations on the fly. Accordingly, Membership Management related capabilities are ideally hosted by the WS-Gateway.

As opposed to this, we foresee the lifecycle capabilities and in particular the Business Process Management to be hosted by a different machine to allow for steering the WS-Gateway in a more stable way (i.e. the WS-Gateway may be easily replaced due to failures). Note that this does *not necessarily* imply that the WS-Gateway and the VO Manager Service are located in different infrastructures.

- Trust & Security:

The particular issue about Trust & Security in the eLearning scenario consists in the WS-Gateway’s capabilities to issue tokens, i.e. to host the VO Security Token Service. Furthermore, it has to administer at least two types of access restrictions: (1) VO specific ones to allow for communication between VO participants and (2) EN specific ones to grant access to the user information (cf. above). Furthermore, it is the only service that the user is allowed to access directly.

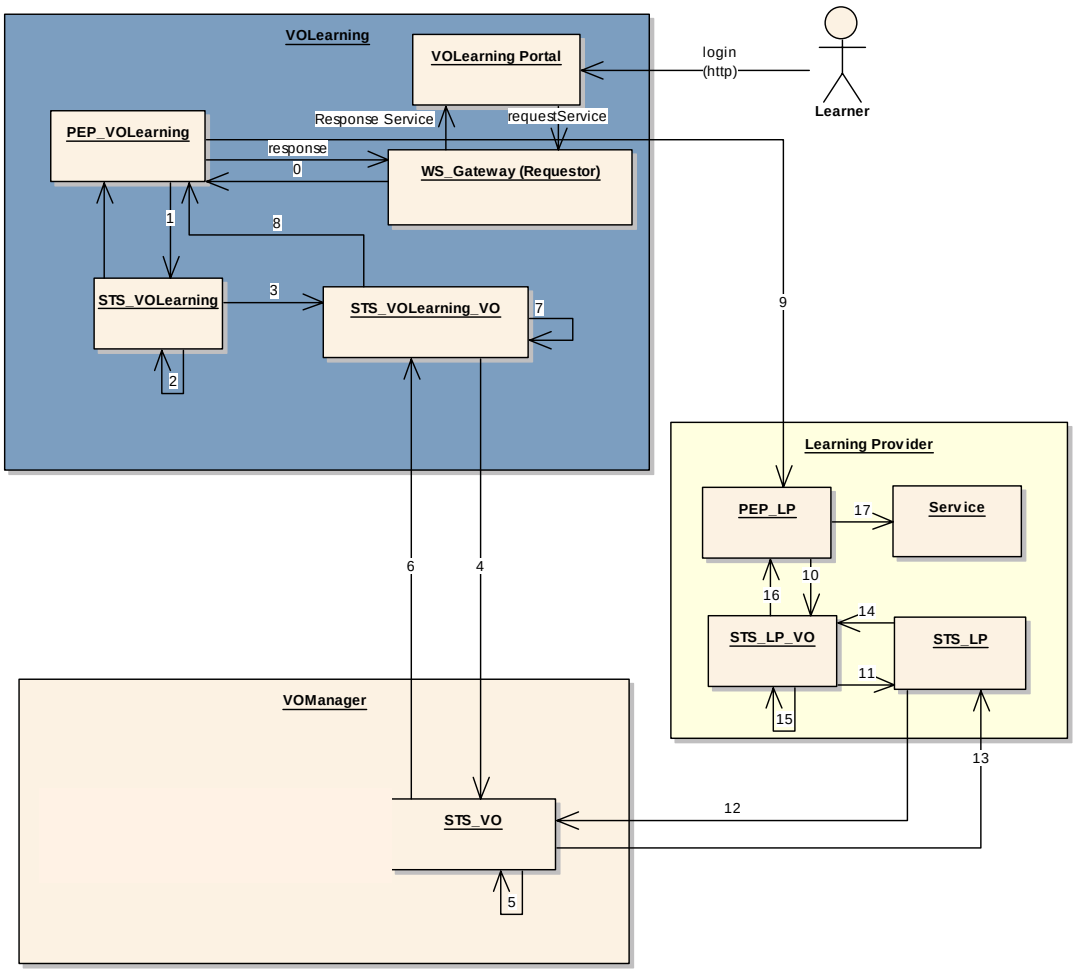


Figure 29: Deployment of Security Token Services in the eLearning scenario

- SLA Management:

The main SLA parameters to be maintained by the Learning Providers in the VO are related to response time: slow or non-reacting services should be replaced immediately, so as to ensure that the learner always gets served as quickly as possible. This poses an issue on retrieving the status information: the service provider will not want to be held responsible for network failures, but only for delays directly caused by his infrastructure, so that the SLA will primarily supervise response time *local to* the Learning Provider. However, since for the user response time *including* network delays are of interest and the VO Manager needs to replace service providers due to network failures (if even only contemporarily), response time information needs also be measured by the WS-Gateway. The main difference between the two methods consists mainly in the responsibility for failure and highly depends on the respective contracts[24].

In the given case, we assume that both Service Provider and WS-Gateway measure response time, that both values trigger replacement of the service once crossing a threshold, but that only for the Service Provider value, this is considered an SLA violation.

[24] Note that the service provider may have a contract with his internet provider that makes him responsible for failures – this is similar to the case of subcontracting and shared responsibilities.

- BP Management:

As noted, the collaboration description in case of the eLearning scenario is really the training path for the student. As such, the VO Manager Service enacts the coordination capability by specifying when the student's requests are forwarded to which Learning Provider, respectively which Learning Provider can communicate with the student. To achieve this, the VO Manager collects progress information from the Learning Providers and advances the workflow accordingly.

Since the eLearning VO is generally very dynamic with respect to its participants, the configuration and setup of service providers needs to be fast and efficient. One way of dealing with this issue consists in the outsourced gateway for VO Management that enables quick message redirection.

Furthermore, the list of alternative service providers as generated during discovery will be maintained for later reference. Since the operational phase of each participant is rather short in the eLearning context, most alternative providers will still be available at the time of need.

Finally, since the SLA related requirements for a service are quite straight-forward (response-time lower than a specific threshold, whereas the possible thresholds may be fixed), the SLA may be verified prior to actual usage of the respective resource without fearing that the according availability will alter too much. This poses no problems for either provider or customer, even if the respective service will not be used at all.

It is worth mentioning that service providers *may* be replaced only temporarily, depending on the underlying contract, i.e. if the network connection fails temporarily, the service provider may want to be re-introduced into the VO, once it is back on-line. Accordingly, this opens up the possibility for "adaptive" services that act as "vaulters" for temporarily unavailable resource. Such "vaulters" may offer only extracts of some lessons but potentially for higher pricing and only for a limited time.

V.2.b Business Models

The basic business model underlying the AS testbed may be regarded as a "two-layer model", strictly following the distinction between an "Enterprise Network" and the actual VO, and the according implications from the legal perspective, i.e. the EN and VO contract:

As already mentioned, we can distinguish three main actor types in the scenario, namely the Student(s), the Course or Learning Resource Providers and the Portal Provider or VO Manager. Alongside these, Trusted Third Parties act as management and infrastructure support.

In most cases, the Learning Resource Providers (LRP) will not encapsulate any further resources or outsource capabilities to other Service Providers, so that no subcontracting needs to be considered. In fact the AS scenario is a very good example for a flat hierarchy where actually all participants not only enter contractual obligations with a VO Manager like instance (the Portal Provider) but also mainly interact with the latter so that responsibility and collaborative relationship correspond perfectly (cf. section II.1).

Each LRP will enter such a form of contractual relationship with joining the eLearning Network, i.e. a kind of Enterprise Network. As such, the VO Manager not only acts as endpoint of the VO contracts but also actively participates in the underlying EN contracts. This does not necessary hold true in all cases though, as we will discuss below.

Though such an underlying EN contract does not necessarily imply that the providers originally *come* from the same consortium, but they implicitly *form* a kind of consortium (see Figure 30).

The actual Virtual Organisation will come about with (and for) each student as he / she enters a contract with the Portal Provider. As opposed to the CE scenario, the EN contract already defines most of the details regarding service provisioning, including the (or a set of) typical SLAs. Since the AS testbed addresses in particular fast creation and adaptation of VOs, this concept will allow that the actual VOs need not spend time on long negotiation and resource adaptation. Instead, the customer (student) requirements are met by according selection of the right providers (that have already entered a contract binding).

As a side note, one may also regard what we consider Enterprise Network here as the actual VO and the individual VOs per student as according adaptations – this does not directly affect the setup or usage of the framework though as, for security reasons, these adaptations must be treated as the individual federations from the middleware point of view.

We have to keep in mind that the testbed is a special use case of VOs in the sense that we need not assume competition between different VOs within the same Enterprise Network. In other words the EN here is particular to one eLearning provider (here Atos Origin with Metacampus).

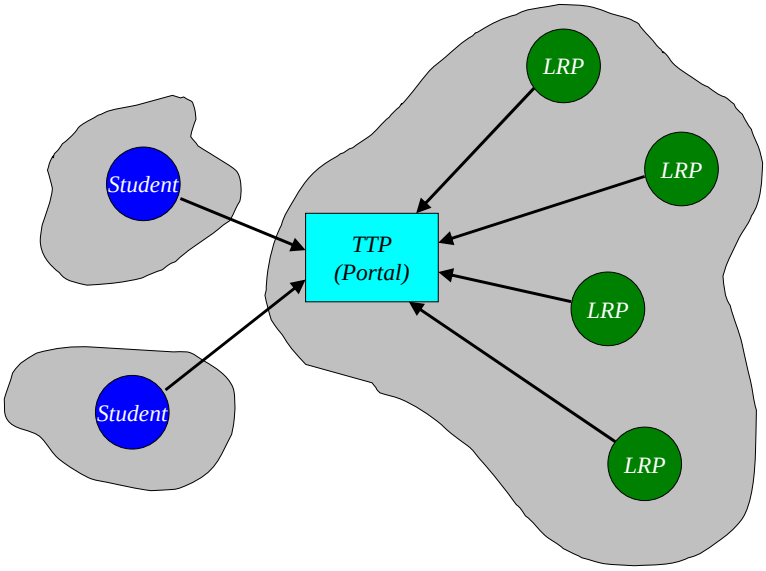


Figure 30: Current business model of the AS scenario

In the straight-forward approach the (one) portal provider obviously poses a bottle neck to many students' requests. As such, the figure above should not be misinterpreted to leave the impression that only one such provider is foresee – instead, with the multiple VO

concept, any amount of service providers may act as portal “interfaces”. In the simplest case, each Portal Provider implicitly acts as a VO Manager to the VO.

VI Methodology

Any approach towards implementing the TrustCoM framework as described within this document must come to the point where the actual data sets per component and how they relate to each other within the framework need to be modelled. Whilst this information may be implicitly derived from the discussions in previous sections, in particular from the relationship models (section III) and the subsystem architectures as defined in Appendix B to this document, this section is also partially derived from the knowledge gained in the implementation efforts during the project lifetime.

As opposed to the more concrete definitions used in the implementation efforts and the actual data profile (Appendix A), this section focuses on the more conceptual issues, i.e. what kind of information is required for the according component and for what reason. As such, this may cover issues not (yet) directly addressed in the development efforts.

Such information may serve three main purposes:

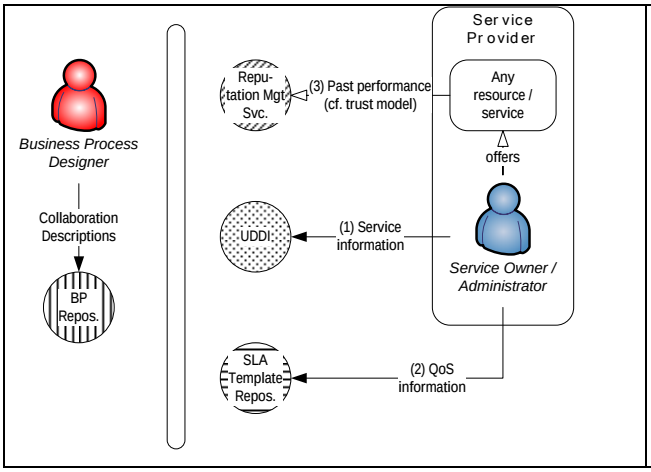
1. to allow SP hosts to create their own *supplementing* components and have them make use of the data structures available in and from the TrustCoM framework.
2. to allow Service Providers to create their own components *replacing* TrustCoM components and yet still being able to integrate them into the framework.
3. to allow future projects to take up the work from a concrete point where functionalities to be extended may be identified more easily.

VI.1 VO Management

The management of the VO follows the operation of the VO lifecycle described in section 1.3 above.

1) Formation of an Enterprise Network (Preparation)

A user representing an organisation will use the user interface on the VO Management toolkit to register details of that organisation in the UDDI registry and will accept the terms of the EN contract as defined in section 1.6a.

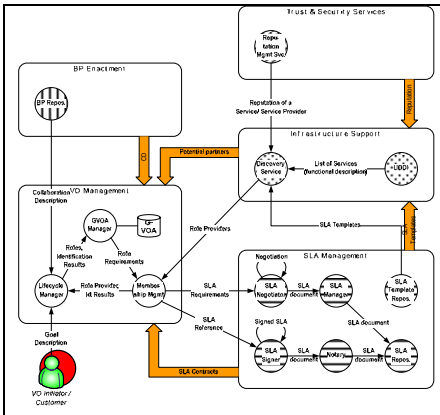


-The Service provider becomes EN Member: agrees to the **EN Agreement** and advertises their services by describing them in the UDDI registry.

- Legally this constitutes an ”Invitation to treat”

2) Establishment of a Virtual Organisation (Identification and Formation)

A VO initiator representing an organisation will use the user interface on the VO Management toolkit to initiate the creation of a VO by stating its objective, defining the Collaboration Definition (as defined in section VI.2.a (1)), define the access control policies and chose the appropriate details required for the GVOA sections on the duration of the contract, and conflict resolution. From the Collaboration Definition the definitions of the roles in the collaboration are extracted by the VO Membership Manager, and matched with UDDI registry entries for candidate VO members, as well as their reputation records from the reputation server. The VO Initiator chooses the best candidates for each VO role, SLA templates are acquired for each service, and offers to join the VO are made.



The invitation to treat in the UDDI directory moves to become an offer from the VO initiator to join a VO, and then following negotiation the offer can be withdrawn or accepted.

After the GVOA has been agreed by partners it will be signed by all parties and VO Members receive proof that the other VO Members signed the GVOA.

If the offers are accepted, all this information is stored in the GVOA registry and a GVOA is generated for signature by all VO members.

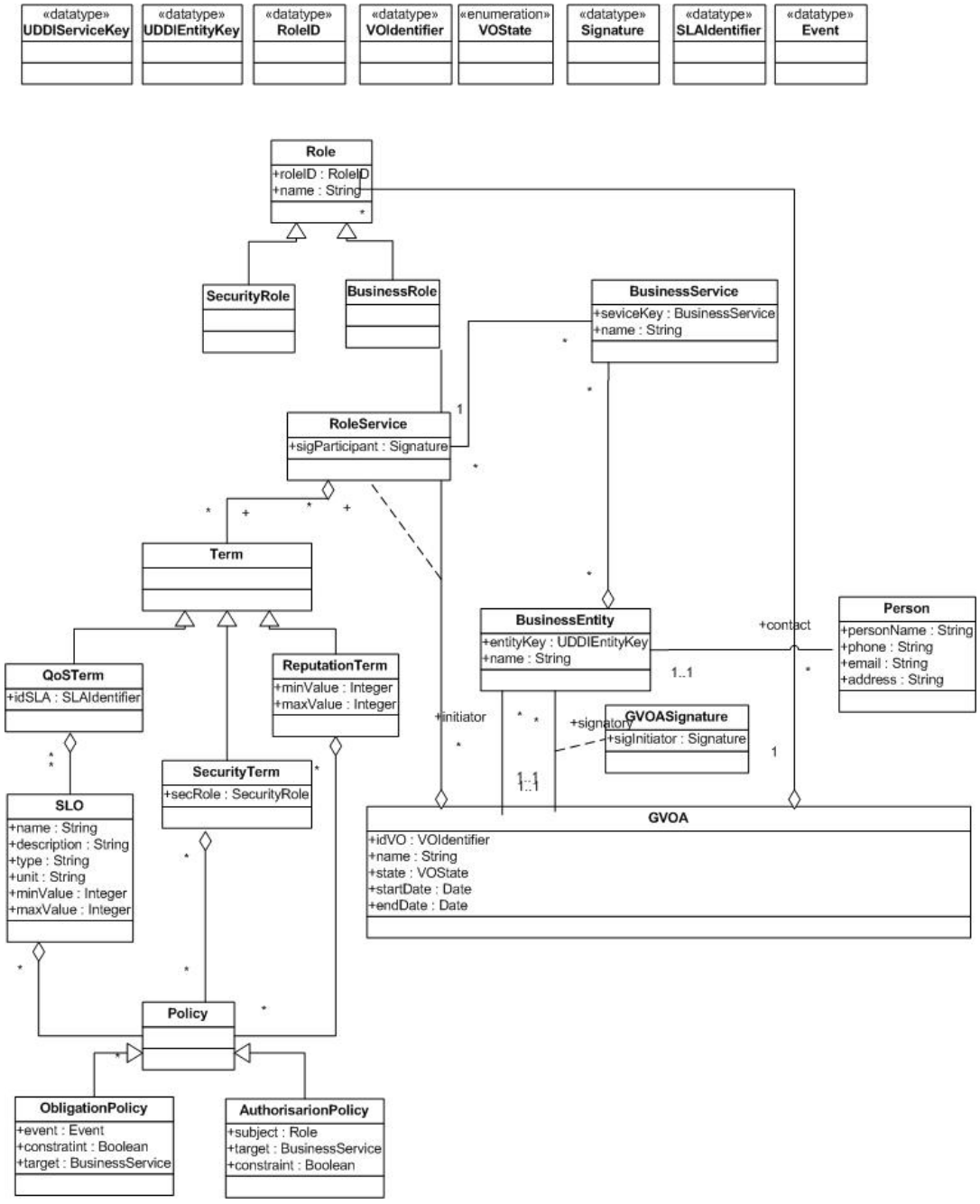


Figure: Class Diagram for GVOA Management

Once the signed GVOA is returned by all collaboration members to the VO Membership Manager the VO can be started. The SLA is sent to the SLA Manager, the access control policies are registered with the security component, the policies concerning actions to be taken for failure to meet SLAs or for breach of access controls are loaded into the Policy Service, and the BP Manager can start the execution of the business process model which drives the normal operation of the VO.

3) Normal operational work

In normal operational work the VO manager has no function since control lies with BP Enactment. The GVOA is enforced by monitoring performance, compare performance to the agreed expectations, and then take management actions when breaches occur. When events trigger breaches in policies in the Policy Server then business processes to manage those policies are enacted. The VO initiator can use the user interface of the VO Management System to inspect the operation of the VO and take actions.

Abnormal behaviour is identified by the monitoring of the business process, SLA, and security components of the system with reference to the GVOA which defines what is expected. Failure to deliver, late delivery, poor quality of service, and security breaches can all be identified as a result of these monitoring activities. The management actions that can be taken the underperforming organisation when these occur include:

- Human intervention and escalation
- Penalty imposition
- Change in business process, to account for delays, or change role between existing partners
- Additional of new partner to VO to undertake all or part of underperformed role
- Removal of underperforming partner
- Replacement of underperforming partner

Each of these management actions is supported by the VO Toolkit.

4) Dissolution of the Virtual Organisation

The VO toolkit supports the dissolution of the VO, while outstanding liabilities continue until final termination by the VO toolkit.

VI.2Business Process Management

As Virtual Organizations are created for a specific objective, there needs to be a means of formally specifying that objective. Within the project the WS-CDL and WS-BPEL specifications were used, but, for the purposes of the framework, we define the components and messaging in a language-independent manner, highlighting the important properties of the subsystem. Business processing languages (e.g. BPEL) provide a means of expressing such an objective, while business process execution engines provide mechanisms for executing an agreed specification. However, as a VO is distributed across multiple administrative domains, the business process is referred to as a “collaborative business process” and must be first modelled from a global/public perspective. The conceptual details of this approach are described in the participant model in section (above). This section describes the information artefacts, components, interfaces and messaging requirements for implementations of business processing in VOs.

VI.2.a Information Artifacts

1) Collaboration Definition (CD)

a specification document that is created by a VO Initiator to model the information and control flow required for collaboration between members of the VO. It includes the roles, relationships, interactions, abstract activities and work-units to be executed by the members in the VO. It is graphically modelled using UML and then translated to a selected, interpretable specification language – WS-CDL was used as the reference specification language. The CD is distributed to all selected members of the VO. Each member is responsible for deriving their private behaviour from this “global” model. The CD may be reused in multiple VOs, such that a reference to an instance is a combination of the VO Identifier and the repository identifier of the CD.

2) BP Pattern

(BP Part for short) a local, private refinement of what a particular member does given it is assigned an abstract activity for its role defined in the CD. BP Parts can be defined in an executable language understood within the domain of a single member. They are never intended to be exchanged across domains.

3) BP View

the derived, limited operational interface that a member provides to its internal services and processes, restricted to agreed members of the VO. This is typically presented as a web service interface, with an associated WSDL. The naming scheme of BP Views is derived from the role names in the CD, such that there is no additional need to agree on the interface name. It is assumed that all members follow this naming scheme when generating their views.

4) Executable BP

the generated business process specification, which can be privately executed with the appropriate BP engine. There is no need for a common naming scheme in this case, as the executable BPs are intended only for internal usage and access. Nevertheless, following a common naming scheme

VI.2.b Subsystem Components and Dependencies

The internal components of the subsystem are listed below, in order of typical usage:

- CD Modeller:

  a component for defining a collaborative business process or collaboration definition using a common, preferably graphical modelling language such as UML (Unified Modelling Language). A CD is a message-based specification of how participants interact, including the control flow that should govern their behavioural interactions. Abstract activities (in WS-CDL referred to as “silent actions”) are used to define points in the model where the internal specific controls and actions are left up to the provider of the functionality and respective participant role.

- CD Repository:

  a file store or data base for maintaining pre-defined collaboration definitions, such that they may be used in multiple VO instances. This repository may be maintained by a VO Initiator or by a central Host that maintains VO specification information.

- CD Parser:

  a mechanism for parsing the CD specification in order to retrieve individual elements such as roles, interactions, activities and other control flow information.

- BP Parts Modeller:

  a component for specifying internal business process (BP) actions and control flow, which correspond to particular abstract activities. These are called “BP Parts” as they are composed to form a fully executable business process.

- CD-to-BP Knowledge Base:

  this is a data base that maintains a mapping between BP Parts and different abstract activities, assuming that there is a common vocabulary for placeholders agreed to in the Enterprise Network.

- CD-to-BP Generator:

  this is the component that executes the algorithm per member in order to compose executable business processes from a high level CD and the mappings between its abstract activities and BP Parts. It also depends on language-specific mappings in order to generate the correct control flow, syntax and semantics. Finally, for every specific execution engine there are additional platform-specific information-sets (i.e. references to external and internal views, communications ports, namespaces etc) required for process deployment, such that the generator also needs to have extensions for this.

- BP Management Service and Engine:

  each member then needs to have an engine that allows the generated executable business process specification to be deployed, started, suspended, stopped and un-deployed. These should only be internally accessible, exposing only the agreed view to the VO.

Most of the messaging in the Business Process Management (BPM) is internal to the subsystem as well as internal to a specific participant domain. The components of the subsystem may however be also available externally as web services or via SQL query interfaces to other subsystems of the TrustCoM framework, depending on the context of usage. There are also certain dependencies that impose constraints on the way the BPM components exchange messages or are interacted with. These are discussed below per subsystem:

- VO Management:

  the VO Management subsystem interacts with the BPM subsystem on a per-phase basis. That is, there are different types of required services of the BPM subsystem dependent on a particular VO’s lifecycle phase.

- Trust and Security Services:

the end point references of generated business processes and the services they compose must be compliant with those validated within the trust and security services subsystem. Without this compliance, the cross-domain business processes will be disabled or special concessions will have to be made for these special types of interactions.

- Discovery Service:

the function of discovery is to ensure that the roles specified in the CD can be fulfilled, based on the registered members of the enterprise network. Discovery also includes the filtering of the list in order to ensure that the reputations and likelihood

- SLA Management:

the BPM components operate unaware of the SLA subsystem. Nevertheless, should an SLA objective not be met, it must be possible to disable the business process, regenerate a valid instance and redeploy it.

- Policy Control:

business processes may be started and stopped due to different policy-based events and conditions.

- EN/VO Infrastructure:

roles and abstract activities must be part of an agreed vocabulary that is available in the EN/VO Infrastructure. All role providers in a business process must be registered members of the EN and provide services that can be defined using the set of keywords specified in the EN's vocabulary. Similarly, if the abstract activities do not conform to a common nomenclature, then there is the chance that their semantics can be misinterpreted, leading to erroneous collaboration and process execution. The area of semantic correctness of processes was however beyond the scope of the framework.

VI.2.c Information exchanges

Information exchanges are described from three perspectives: (i) VO Host (a third party, such as an EN administrator that provides infrastructure services), (ii) VO Initiator and (iii) VO Member

1) Preparation

(i) A VO Host may supply pre-defined CDs in a repository, but makes the listing of keywords and other vocabulary items available to its registrants

(ii) A VO Initiator may maintain its own local repository of pre-defined CDs, ensuring that they are compatible with the keywords and vocabulary established by the Host

(iii) VO Members define their local BP Parts, corresponding to the keywords for which they have registered. These are stored in their local knowledge base.

2) Identification

(i) A VO Host provides a registry for discovering providers of roles specified in a CD, given that the roles are compliant with those in its keyword database.

(ii) A VO Initiator parses the CD and uses the roles to do the respective discovery queries for potential members. The Initiator then sends invitations that include the CD as an attachment to each selected member.

(iii) Candidate VO Members receive invitations for playing specific roles in the CD. They have the option to accept or reject them. If they accept an invitation, they must ensure that they have the respective mappings to the abstract activities included in the CD.

3) Formation

(i) At this stage the selected members for the VO are already stored in a registry maintained by the Host (see VO Management)

(ii) The Initiator also needs to maintain references to which members have been assigned to the roles in the CD

(iii) Each selected VO Member does the following:

a. Parses the CD, creating and initializing the required objects

b. Validate the CD (given the agreed syntactical and semantic standards). The typical validation includes that of attributes, elements, variables and other constraints of the specification language

c. Executes its local CD to BP generation and mapping algorithm, in order to create private processes and make process views available. The following points need to be noted:

- Standard elements of the CD and BP language are first mapped and directly translated
- If the current element cannot be translated directly, there should exist a KB lookup for the complex generation step, where the input is the element that cannot be mapped. This is typical of abstract activities but it this may also be specialized for different types of languages
- If it is still not possible to perform the translation of the elment, then an exception has to be raised and a manual translation correction procedure performed. (We have not made provisions for this in software, and assume that this is resolved offline). Automated means of doing this are beyond the scope of the framework.
- Deployment-specific information types then need to be compiled and included into the BP specification.
- Report back to the “calling process” (or local service handling configuration) that the BPs have been successfully generated and deployed

4) Operation and Evolution

(i) The aim is to reduce the involvement of the VO Host after the formation phase as much as possible. A CD specifies a peer-to-peer protocol in most cases.

(ii) The VO Initiator is responsible for invoking the first process in the CD, such that a record of the end-point-reference of the member assigned this role must be explicitly maintained. There is a special `initVOChorVar` variable included in the CD that indicates this role, such that the initialization interface needs to be generated (see 3, c)

(iii) VO Members execute their private BPs transparently of the overall, global CD

Exception Handling

Another area of BPM is that of exception handling. This is considered a complex topic in distributed systems such as VOs. A concept was developed in TrustCoM for firstly classifying different types of exceptions, namely Trust, Contract and Security (TSC) exceptions, as well as a mechanism for catching, propagating and handling exceptions in a distributed manner, using the concept of compensation. TSC tasks are special types of abstract activities, which must be agreed to by each member and a local implementation provided.

5) Dissolution

(i) Completion of the CD typically coincides with dissolution of the VO.

(ii) The VO Initiator should ensure that the global process specification (the CD) has been executed successfully or to a point where dissolution of the VO is possible

(iii) Each member has the option of undeploying their processes that were created specifically for the VO.

VI.3SLA Management Services

This section describes the information artefacts and data exchanges related to the SLA Management subsystem.

VI.3.a Information Artefacts

1) SLA Document

An ‘SLA Document’ is a data structure which contains information about a service level agreement between two parties. This information includes the parties’ identities, the location of the service agreed upon, the definition of monitors and metrics used to quantify relevant aspects of the service, and the definition of QoS objectives using the metrics defined.

Every SLA Document has a globally unique identifier (unique across VOs and time), and it is stored in the SLA Repository. An SLA Document can be recovered from the repository by means of its identifier.

2) SLA Template

An ‘SLA Template’ is a data structure which defines a template for the creation of SLA documents. It contains information describing the constraints under which a certain party offers to provide a service. It is used during the negotiation of a final SLA document which will regulate the provision of the service within a VO.

SLA Templates are stored in the SLA Template Repository.

3) SLA Reference

An ‘SLA Reference’ is a data structure which identifies a particular SLA Document. The globally unique identifier of an SLA Document is used as its SLA Reference.

4) SLA Status

The ‘SLA Status’ is a data structure which contains information reported by the monitors of a service, in accordance with the corresponding SLA agreement. A monitor of a service reports values corresponding to specific metrics defined in the SLA document for the service. These values are required for the evaluation of QoS objectives also described by the SLA Document.

5) SLA Notification

The ‘SLA Notification’ data structure contains information on the violation or fulfilment of QoS objectives, reported by an SLA Evaluator in accordance with a corresponding SLA Document. SLA Notifications are distributed to other subsystems, and may be stored for later reference in the SLA Performance Log.

VI.3.b Information exchanges

1) Preparation

A member of the Enterprise Network registers the services it is willing to offer to potential VOs by listing their descriptions in the Service Registry. These descriptions have associated SLA templates that are stored in the SLA Template Repository.

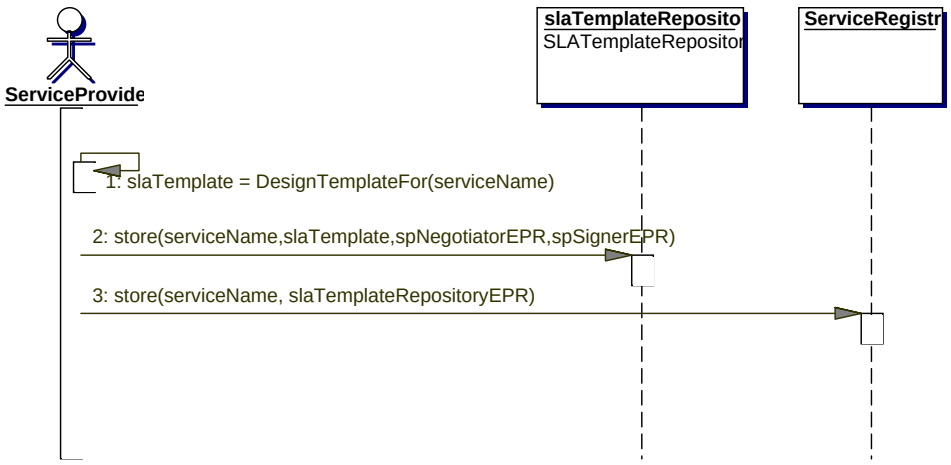


Figure 31: Register an SLA template

2) Identification

The VO Manager will use the services of the Discovery service to search for services to fulfil the requirements of the collaboration definition, including QoS requirements.

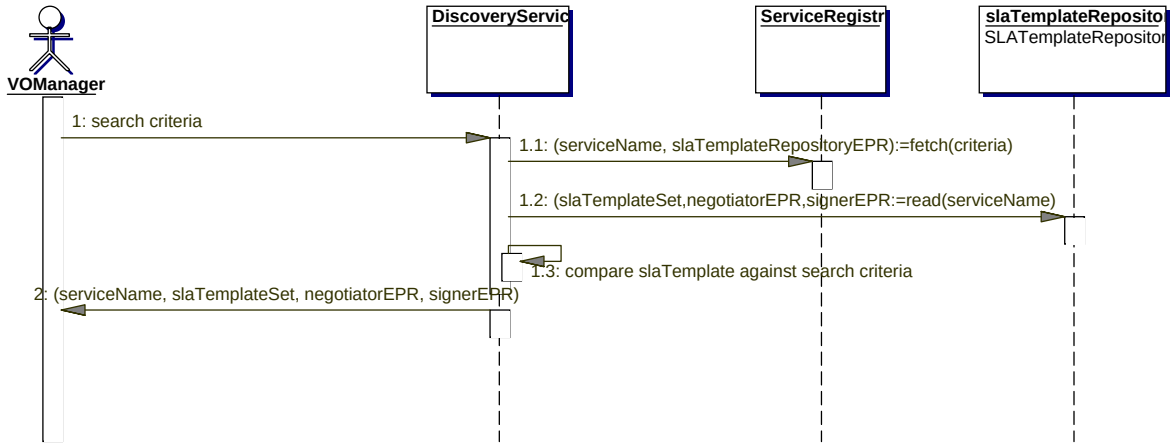


Figure 32: Discovery service using QoS requirements

The VO Manager handles a simple version of an SLA negotiation protocol, here restricted to a single round where the offer, made by the service consumer based on the SLA template, is either accepted or rejected on the spot by the service provider.

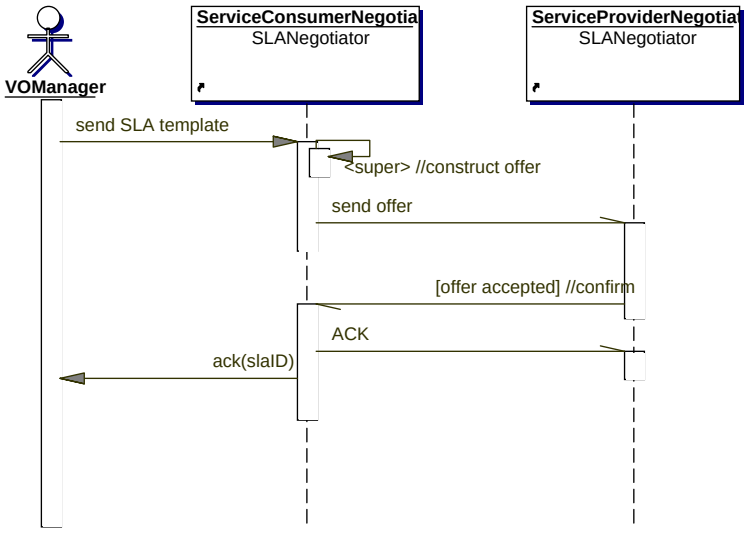


Figure 33: Single-Round SLA Negotiation

3) Formation

The VO Manager initiates an SLA signing protocol. In the simple case depicted below, the protocol involves a Notary that first collects all signatures, verifies them and then distributes the signed contracts among the signatories. The Notary communicates the result of the negotiation to the VO Manager, and stores the signed contract in the SLA Repository.

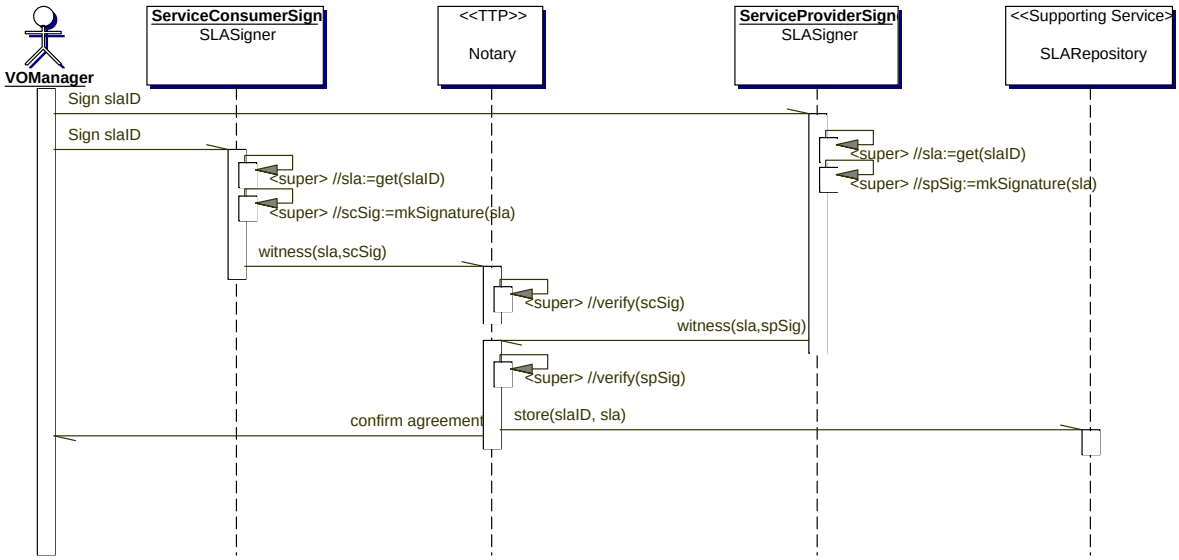


Figure 34: Signing and Storage of SLA Documents

Configuration of Evaluators and Monitors

The VO Manager uses the VO SLA Manager to configure Evaluators and SLA Monitors in order to monitor SLA performance. This task is completed with the help of the Partner SLA Manager when there is a need to configure monitors and evaluators that are internal to an organization. The Partner VO Manager also maps “public” SLA parameters to internal configuration information.

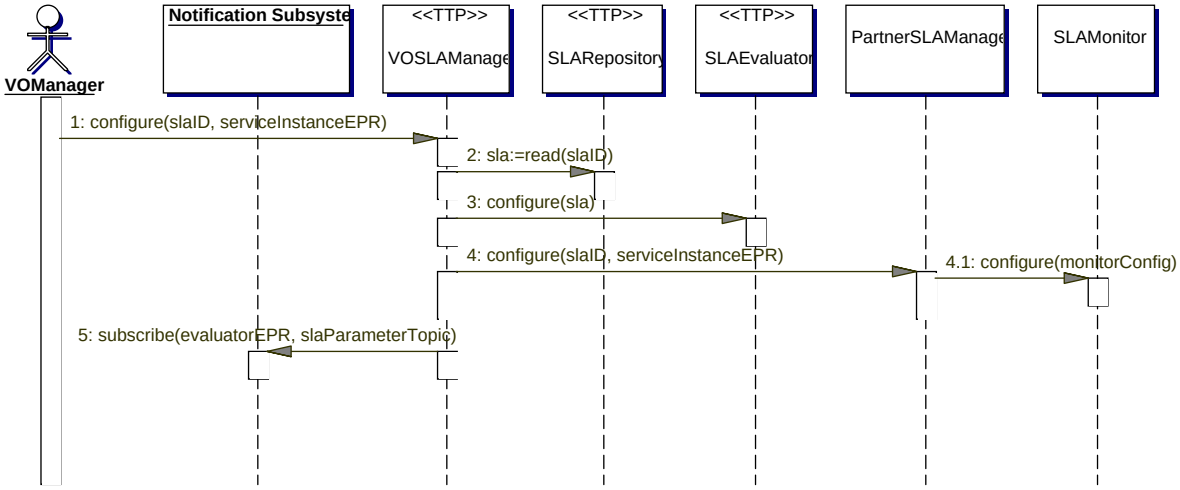


Figure 35: Configuration of SLA Monitors and Evaluators

4) Operation and Evolution

The SLA monitors observe the execution of a service, host process or even business process, according to the configuration provided during the formation phase, and compute SLA parameters according to the metrics defined in the corresponding SLA. These

parameters, representing the status of the service, are communicated upon the occurrence of events (including time events).

Notifications of SLA performance are generated by the SLA Evaluator and channelled to the Notification component to be distributed to the receivers that were subscribed in the Formation phase (e.g. to the Policy subsystem).

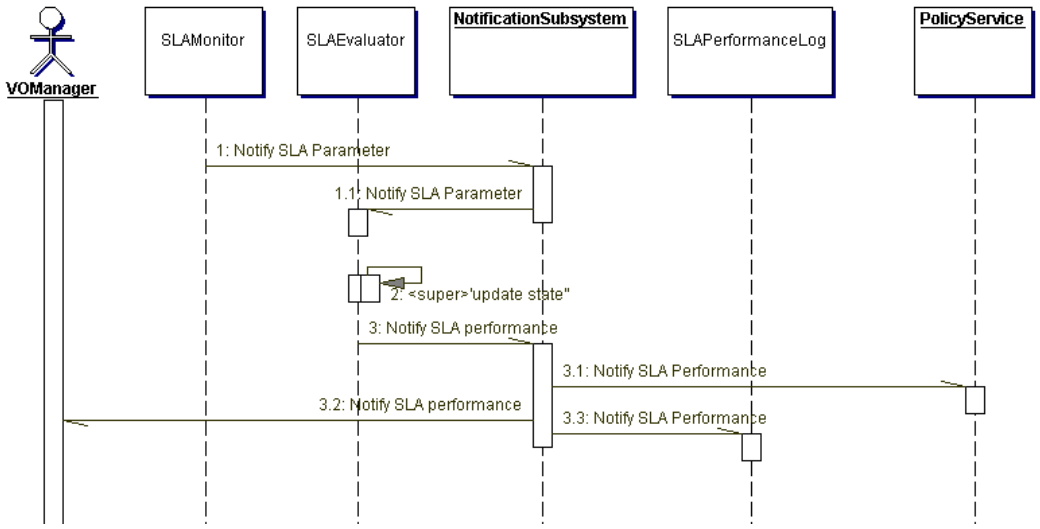


Figure 36: SLA Monitoring

VI.4Trust & Security Services

VI.4.a Security Token Service (STS) related

This section describes the information artefacts and data exchanges that TrustCoM components exchange related to security token services.

1) Information artifacts

This section lists the security-related information artifacts that TrustCoM uses.

Business cards

A 'business card' is a data structure that contains information about a company. The data contained in a business card is stored in the UDDI repository. When a company joins the enterprise network, that data is uploaded to the UDDI repository. A business card contains the identifiers and cryptographic information that is necessary to identify a partner inside a VO.

A business card MUST contain the following information:

- *The company's STS' public endpoint reference*, which can be used to directly contact an STS in order to validate, renew or revoke security tokens. In the currently implemented interaction model, that public endpoint is not contacted cross-organizationally, but we consider enhanced collaboration models where that information would be necessary.

- *The company's STS' security token(s)/cryptographic keys.* An STS may have multiple security tokens, such as X509 certificates or public keys. Each STS must have a security token that is used to digitally sign issued tokens (data origin authentication). Another token may be used for confidentiality protection.

A business card MAY contain additional identification information, such as

- the company's UDDI business entity key,
- the company's homepage link,
- the company's legal entity name,
- ..., etc.

VO Identifiers

Each virtual organisation has a unique identifier, called the 'VO ID'. A VO ID (sometimes also called 'federation UUID') is a UUID-compliant identifier that the VO Management generates when establishing a VO.

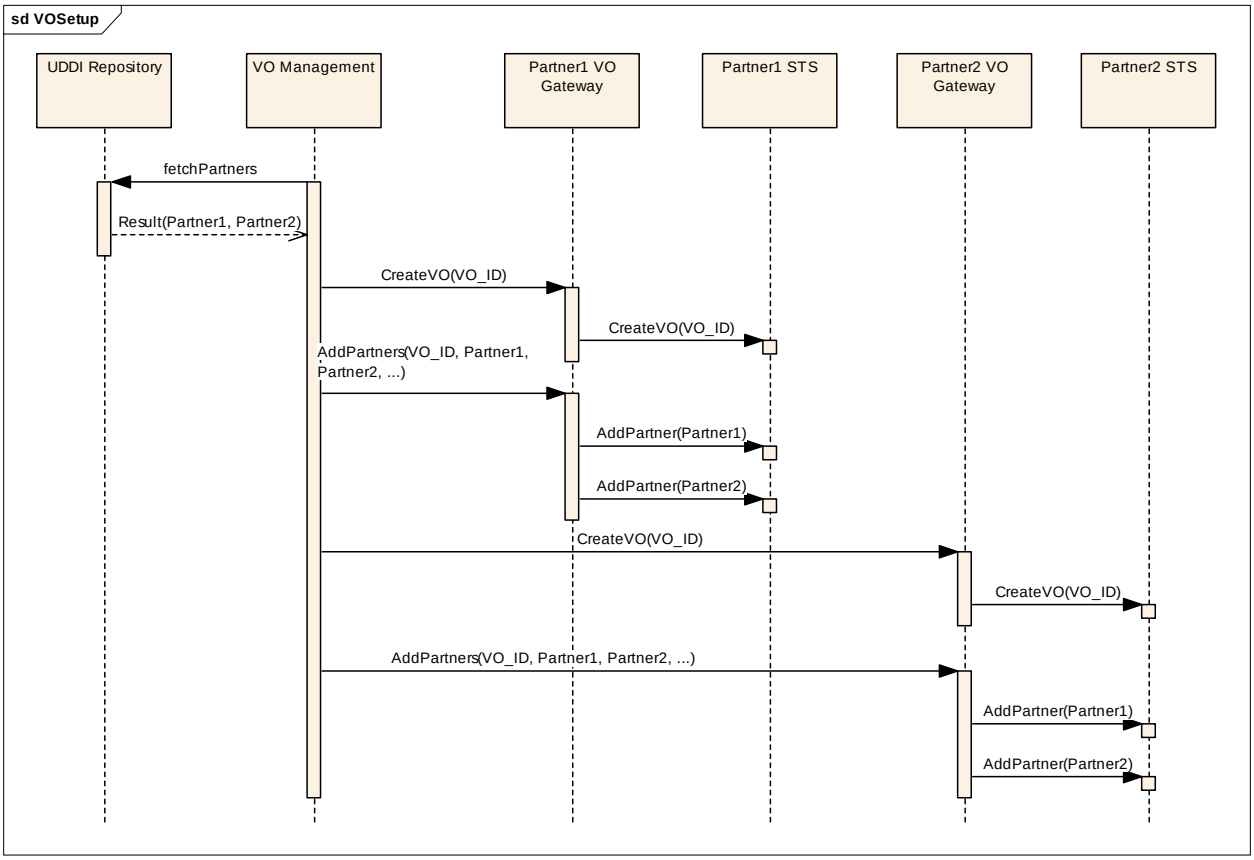
Cross-organizational security tokens

For cross-organizational service invocations, TrustCoM security tokens as defined in the TrustCoM SAML profile section. These security tokens are SAML 1.1 assertions that enable peer entity authentication, authorization and confidentiality protection for cross-organizational message exchanges.

2) Information exchanges

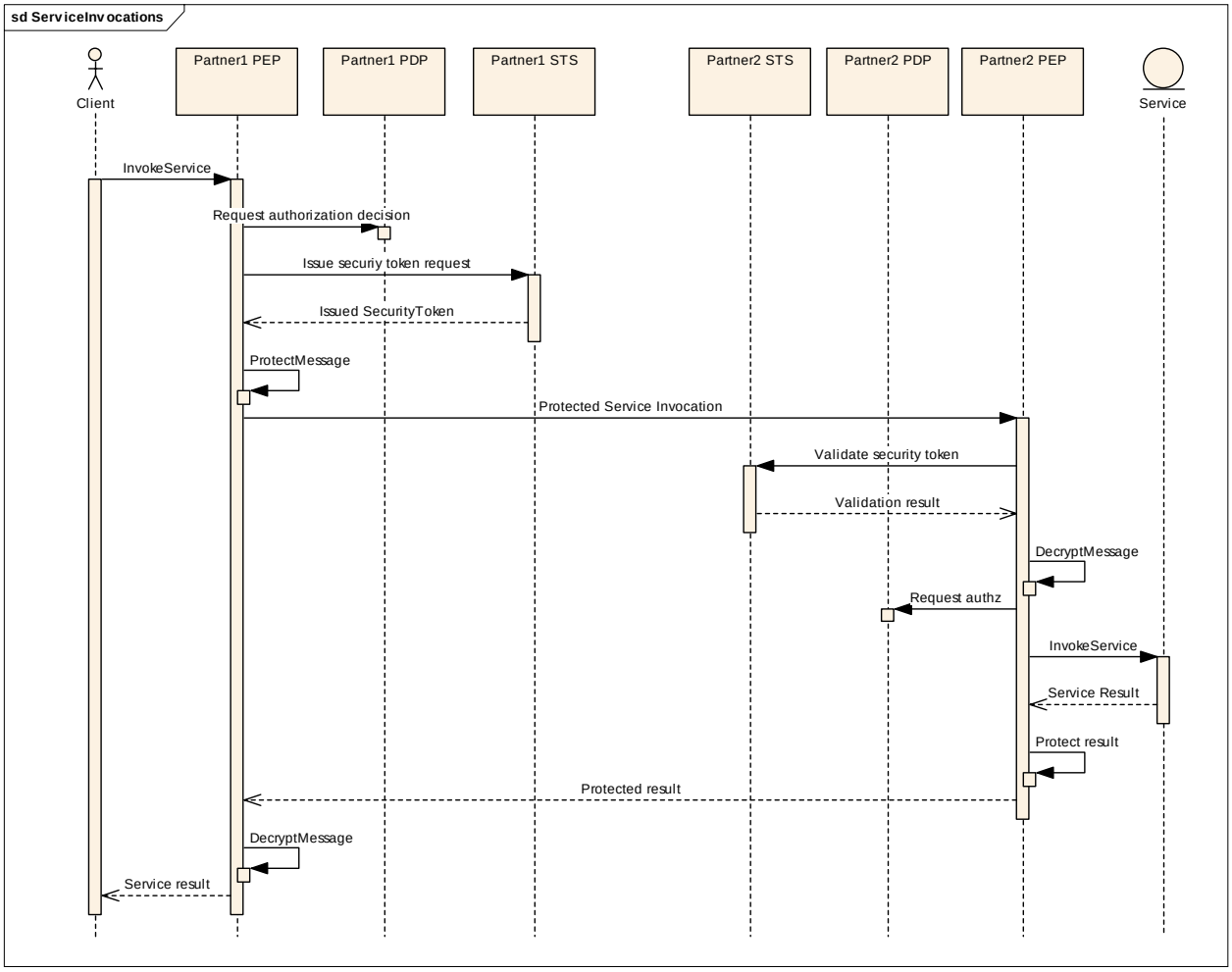
VO Establishment

When the VO Manager establishes the VO, it selects all partners (after SLA negotiation, etc). Once the VO member/partner selection is done, the VO Management fetches the partner's business cards from the UDDI repository (if not done previously). The VO Management notifies each partner's 'VO Gateway' about the establishment of the VO and uploads the business cards of the VO partners. Each VO gateway then configures the partner's STS accordingly.



Service invocations

When a client invokes a service, the client's message is intercepted by the client's PEP. That PEP fetches a security token, protects the message with that token and sends the message to the other organization.



VI.4.b Reputation Services related

This section describes the information artefacts and data exchanges that TrustCoM components exchange related to reputation services.

1) Information artifacts

This section lists the reputation-related information artifacts that TrustCoM uses.

Reputation values

A 'Reputation value' is a data structure that contains the trustworthiness of a particular VO participant, or service. This data is calculated by the reputation evaluator and stored in the reputation management service. Usually, this is a value between 0 and 1, where lower values denote less trustworthiness, and 0.5 means “no information” available. Other trust metrics MAY be used, e.g., a finite number of discrete trust levels. A reputation value MAY also be composed of a number of values for different contexts, in which these trustworthiness rating apply.

Reputation change notifications

A 'reputation change notification' is a data structure that contains

- the „business key“ of the effected VO partner,

- the new reputation value

Additionally it MAY contain information like

- the previous reputation value, or the difference between the previous and the current reputation value,
- a description of the reason for the change of the reputation value.

2) Information exchanges

Discovery

When searching for adequate services, the discovery service asks the Reputation Management service to provide reputation values for particular services. These reputation values are then used by the discovery service to filter those services which reputation values are below a specified threshold.

Operation

On receiving SLA Notifications, the Reputation Evaluator calculates the new reputation and in return sends out reputation change notifications (which can then be used, e.g., by the policy services).

Dissolution

The Reputation Evaluator may request archived SLA Notifications from the SLA performance log and archived messages from the secure audit log to calculate a „final“ reputation value that is then stored in the reputation management service.

VI.4.c Secure Audit related

This section describes the information artefacts and data exchanges that TrustCoM components exchange related to the Secure Audit Web Service (SAWS).

1) Information artefacts

SAWS Log Data Item

A 'SAWS Log Data Item' is an opaque byte sequence consisting of the information that someone wants to have logged undeniably.

SAWS Log Entry

A 'SAWS Log Entry' is a data structure that contains

- an identifier,
- a timestamp,
- a SAWS Log Data item,

describing „who said what at which point in time“.

2) Information exchanges

PEP's may use the SAWS to log in- and outgoing messages as SAWS Log Data Items during operation of the VO. Anyone (who has the necessary authentication information), e.g., administrators, may query the SAWS for SAWS Log Entries of a particular VO partner.

VI.5Policy Control

This section describes the information artefacts and data exchanges that relate to the specification, deployment and enforcement of policies in a VO. The TrustCoM framework is policy-driven because this enables the framework to dynamically adapt to changing situations such as SLA violations but also because this enables the framework to be tailored to the characteristics of the VO and of the application domain. For example, a collaborative engineering VO in which relationships tend to be of longer duration may respond differently to a loss of reputation than an aggregated services scenario in which service aggregation happens with shorter life-spans. Similarly, the deployment of the policy components may change according to the needs of the VO. Thus, multiple policy services may be used in larger VOs or VOs in which procedures at both central and local level are governed by policy whilst a single service may be used in medium size VOs in which only collaborative behaviour is policy driven.

This section implicitly also describes a methodology it terms of the sequence of steps that are typical for establishing and enforcing the policies of a VO. Although these are not mandatory and variations can be encountered we expect that in most cases similar steps will be followed.

1) Information Artefacts

Policies

We are primarily concerned with two types of policies: *obligation* policies (also sometimes called adaptation policies) which are specified in terms of event-condition-action rules and define what changes need to happen in response to events occurring, and *access control policies* that include authorisation and delegation policies and which define which subjects or principals have permission to access specific services under given conditions. Both policy types are expressed in an XML notation, which is described in detail in the documentation of the policy-service and the PDP components. Briefly, authorisation policies follow the XACML 1.1 standard with extensions for expressing delegation policies. We have had to define our own encoding for obligation policies as these are not covered adequately by any existing standard. An “event-condition-action” rule encoding was chosen because it promotes decoupling of the services in the VO, dynamic extension in terms of both administrative services and policies in the VO and permits multiple services to react concurrently to the same event.

Policies define the procedures for the VO functioning and as such need to be agreed by all participants before the formation of the VO. This is required because negotiation of policy content requires human intervention and cannot be automated beyond trivial examples. Although it is possible to derive skeleton authorisation policies from the business process requirements for the VO, human intervention is required in order to both review the policies and specify constraints on the authorisation e.g., specific times of day.

Notifications

Notifications are the events that trigger the obligation policies and are described in more detail in section VI.6.b and Appendix B to this document. An event may comprise attributes that can be used in the evaluation of the conditions or as parameters in the actions of the policy.

Policy Actions

Policy actions are typically web-service invocations that are made on administrative services within the VO such as VO management, reputation, or PEPs. However, the policy service also permits the dynamic loading of adapter objects in order to accommodate other services and legacy applications within the VO framework. Each policy defines the sequence of invocation that must be made and on which target services.

Authorisation Request

An ‘Authorization Request’ is a data structure in the form of an XACML request context. The service requesting an authorization decision will fill in the subject, resource, action and environment attributes of the request context.

Authorization Response

An ‘Authorization Response’ is a data structure in the form of an XACML response context. The response context contains either a ‘permit’ or ‘deny’ decision and/or error information related to the processing of an authorization request.

2) Information Exchanges

Formation and Evolution

During the formation and the evolution phase of the VO, policies are loaded into the policy service(s) and into the policy decision points. Policies are typically derived from the terms and conditions of the VO Agreement, although it is expected that within the context of an Enterprise Network VOs will follow similar administrative/management procedures and therefore specifications can be derived from common templates and formats. Although it would be desirable to automate the transformation of GVOA terms and conditions into policies that can be enforced by the system this is not feasible in the general case. Partial automation techniques for policy refinement in VO environments was not within the scope of the TrustCoM project and remains an item which will require further investigation at the end of the project.

Figure 37 below shows the typical sequence of invocations that occurs for deploying policies during the formation and evolution phases of the VO. Either the VOManager or the GVOA may deploy the policies to a policy service. The policy service then deploys authorisation policies to the PDPs. Conceptually it does not matter whether the policies are loaded to the policy service by the VOManager component or the GVOA component; this depends on whether the GVOA acts as a “passive” repository of information or whether it also implements coordination functions for distribution of content. For larger VOs where multiple policy services are required, policies can also be deployed between the policy services.

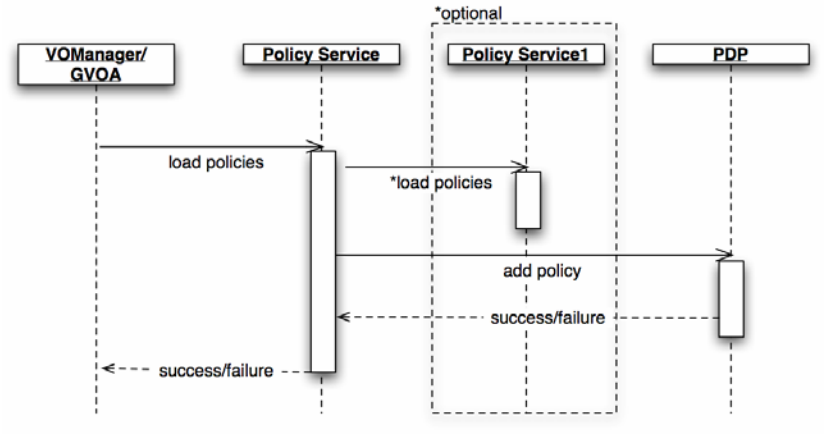


Figure 37: Policy Deployment

This diagram assumes that the membership of the VO is known, partners have been assigned to their VO roles and the PDP for the services pertaining to those roles is known. However, membership of the VO may change during the VO operation or partners may be assigned to roles at different moments in time. Such changes would be published as a notification by the notification broker. When received by the Policy Service such a notification can trigger the un-loading of policies from the PDP removed and deployment of the policies to the new PDP. This is represented in Figure 38 below. Note that this may equally occur during the operation phase of the VO.

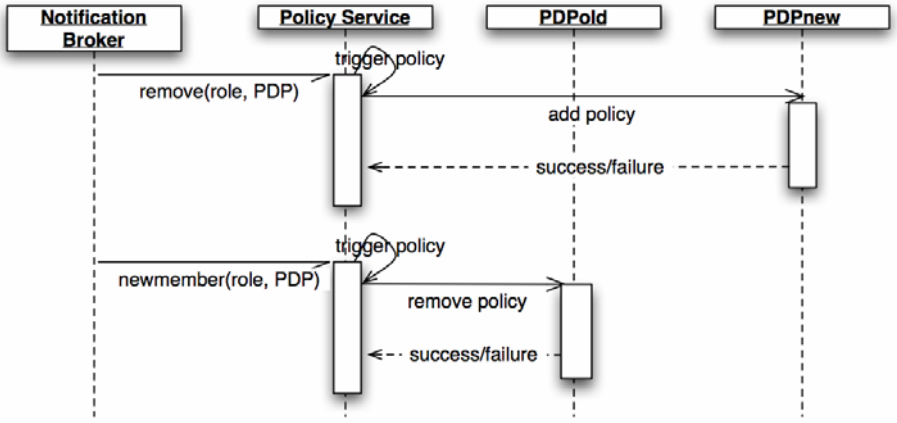


Figure 38: Event-based deployment

VO Operation

When a consumer wants to invoke a service, the Policy Enforcement Point (PEP) will fill in the subject, resource, action and environment attributes of an ‘Authorization Request’. The PDP will then evaluate the query in the XACML engine based on its available policies, and return an ‘Authorization Response’ as a result (Figure 39).

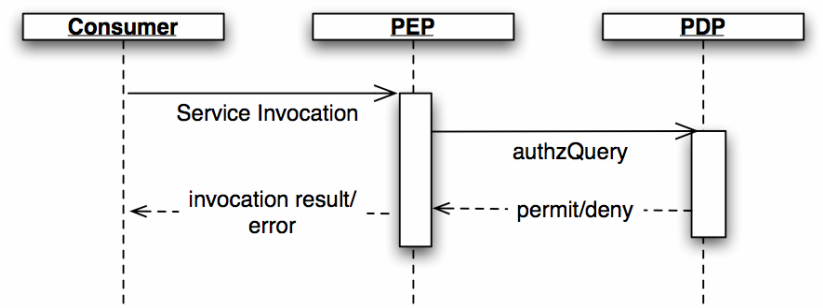


Figure 39: PDP access authorization query

Whenever an obligation policy is loaded in the policy service, the policy service contacts the notification broker and subscribes to receive the 'notifications' that are specified as part of the policy (if it is not already subscribed to those notifications). Several components may generate 'notifications' within a VO such as the SLAEvaluator, the Reputation service, the VOManager, and the PEPs. For each notification received the policy service matches the notification received against the policies that it has and evaluates the constraints specified for those policies. If the constraints evaluate to 'true' the policy 'actions' are executed Figure 40.

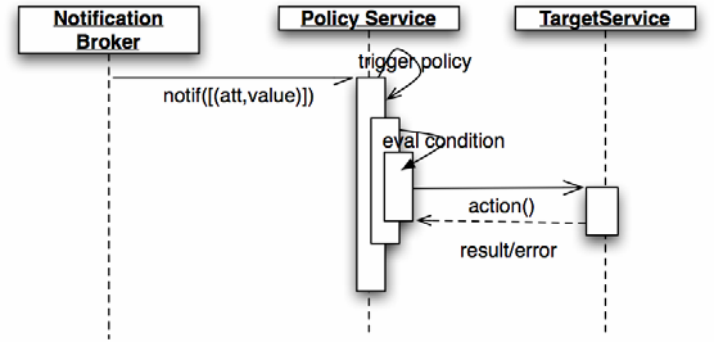


Figure 40: Adaptation actions through obligation policies

The target service in the figure above can be any of the services in the VO infrastructure. Typical policies in a VO include:

- Configuration of services when new members are added to the VO.
- Reconfiguration of services and in particular the PEP message handlers when services or the configuration of services changes within the VO.
- Removal of a member or triggering administrative procedures which lead to a process of removal in case of persistent violation of SLAs or significant changes in the reputation of partners.
- Loading or unloading of policies from both the policy service(s) and the PDPs when the membership of the VO or the terms and conditions governing the functioning of the VO.
- Performing "clean-up" procedures for VO dissolution.

Although obligation policies have been considered for the management and adaptation of the VO framework, they can also be used for application purposes.

VI.6EN/VO Infrastructure

VI.6.a Gateway related

The central capability of TrustCoM EN/VO infrastructure is a business-to-business Gateway (GW).

The Gateway Component allows an enterprise to expose different capabilities as web services in a secure, dynamic, and virtualized manner. The virtualization is guaranteed via the creation and management of service instances which contain infrastructure-specific configuration including security parameters. High level architecture of the GW is shown in Figure 41.

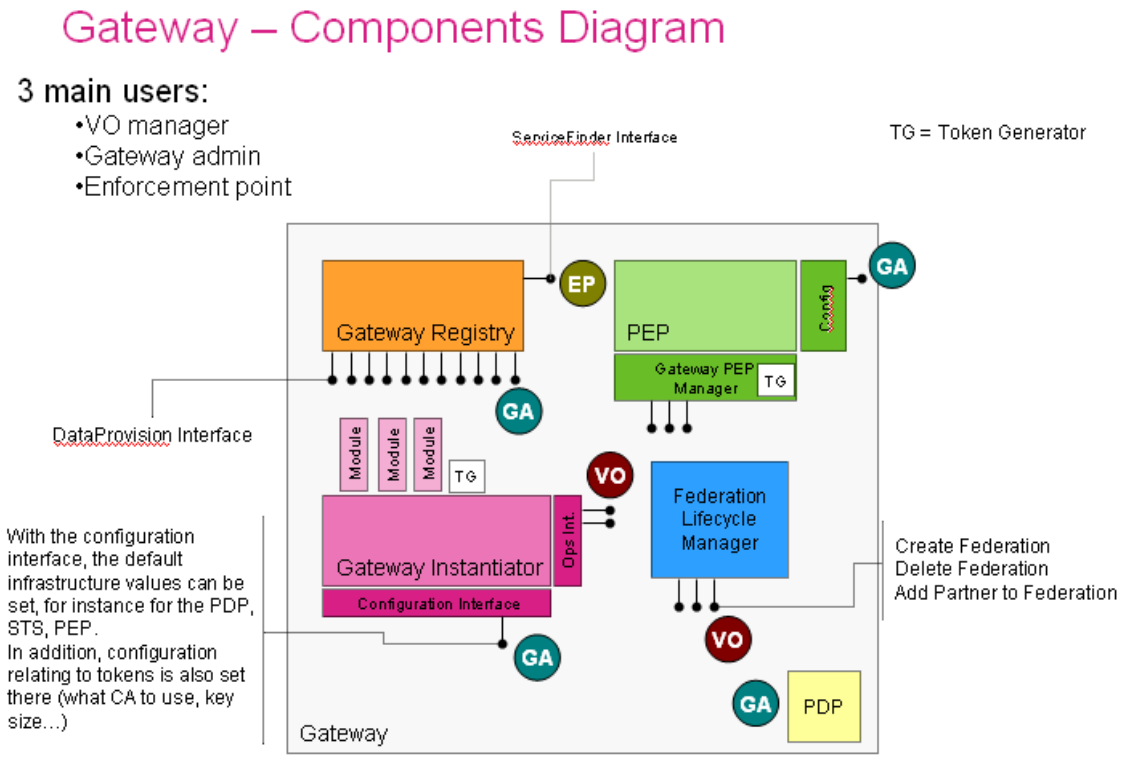


Figure 41: High-level Architecture of TrustCoM EN/VO Gateway

The gateway is made up of two type of services: core services (providing the basic infrastructure functionalities), and optional services that bring additional features (e.g. a specific type of security).

Brief description for each of the GW services is given below:

1) The Gateway Registry.

The gateway registry is the repository of the local information, where Gateway administrators can define and store capabilities[25]. As the capabilities are instantiated and evolve in a given context, or federation, this information is also stored in the registry, including the profiles and federation templates that define which and how infrastructure services should be used. Finally, the registry stores information relating to the partners that take part in a federation; this includes partner's company information, its business card (stored alongside partner information) and business key.

2) The Token Generator.

The Token Generator is an application that provides the gateway with certificates issued by the gateway CA. These certificates are then used by the policy enforcement point (PEP) to sign the issue / validation request going to the security token service. In addition, the certificate is also embedded in the STS-bound request. The Token Generator API closely mimics the structure of the OpenSSL implementation.

3) The Gateway Federation Lifecycle Manager.

Federations are the context in which service instances evolve. They have a unique identifier and contain a set of business cards representing the parties taking part in the federation. A federation, when initially created, is disabled and cannot be used. The Federation Lifecycle manager allows the gateway administrator to create and manage federations through the following interactions:

- *Federation creation*, which includes three main actions:
    - The federation lifecycle manager determines which profile to apply. The profile identifier can be supplied by the user, or otherwise the default profile is used as specified in the component's configuration.
    - The federation is pushed to the gateway registry.
    - Upon successful registration, the federation is pushed to the appropriate STS via the latter's management interface (as defined in the federation profile applied to the federation being created).
- *Federation deletion*, which includes removing the federation from the registry, and removing the federation from the STS (again via the management endpoint stored in the federation).

4) The Gateway Instantiator.

It allows for service virtualization via the creation and management of service instances, the configuration of the gateway infrastructure on a per-instance basis, allowing for tailored security configuration set at the STS, PEP, and PDP. Input arguments to the instantiation request are: the identifier of the capability to be instantiated, the new logical address (provided by a requestor), and the federation ID. Depending on the case, the requestor may or may not be able to specify claims that define the instance's role in the given federation. The instantiation process includes the following steps:

[25] Capability is an application or service hosted by a given company that usually takes in information, computes it and returns a meaningful result to the invoker.

- Claims management. At the start of the instantiation process, the administrator examines the claims, and can add / remove claims to the request, as well as deny the request altogether.
- Data validity check validates the data input in the request against the existing data in the GW registry, as well as that the specified instance does not already exist.
- Loading of GW gateway configuration; this depends on the federation in use.
- The logical address provided is checked to be coherent with the infrastructure's PEP setting (the PEP operational address must be used in the logical address: same port and same domain).
- A new instance identifier is created.
- Based on the instance identifier, a new X509 certificate is created.
- PEP configuration: the certificate bundled with its private key is pushed to the PEP along with different policies defining the behaviour of the PEP when an application message from / to that instance reaches it. This step is role-dependent. The contents of the policies pushed will depend on whether one is instantiating a client or a service.
- The newly created instance is stored in the local registry.
- The instance is pushed to the STS.
- The instantiation process returns an EPR, as defined by the WS-Addressing. This EPR contains the logical address, the federation identifier, and the partner business key.

The optional components are:

- Security Token Service (cf. section VI.4.a and Appendix B, chapter IV)
- Policy Decision Point (cf. section VI.5 and Appendix B, chapter V)
- Policy Service (cf. section VI.5 and Appendix B, chapter V)
- Policy Enforcement Point (PEP). The PEP contains 3 main components:
    - *The management component* defines how resources are created and managed. It deals with virtualization of resources and the management of the corresponding policies. It maintains an internal database that the enforcement component uses to locate the policy for a given endpoint.
    - *The interceptor component* defines how resources are exposed and messages are intercepted. It is responsible for handling the message on the network. It intercepts the message, strips the encapsulated application message from the transport protocol, feeds it into the enforcement engine and embeds the processed message into a new transport level message and sends this on to the next destination.
    - *The enforcement component* defines how policies are located and messages are transformed. It processes the SOAP message based on the policy it retrieves for this virtualized resource and any other information that can be derived from the state of the virtualized resource or the context of the

message exchange. It sends relevant information about the enforcement process via RMI to the management interface of the virtualized resource which publishes these messages to WS-Notification consumers. Figure 2 shows the relations between the components.

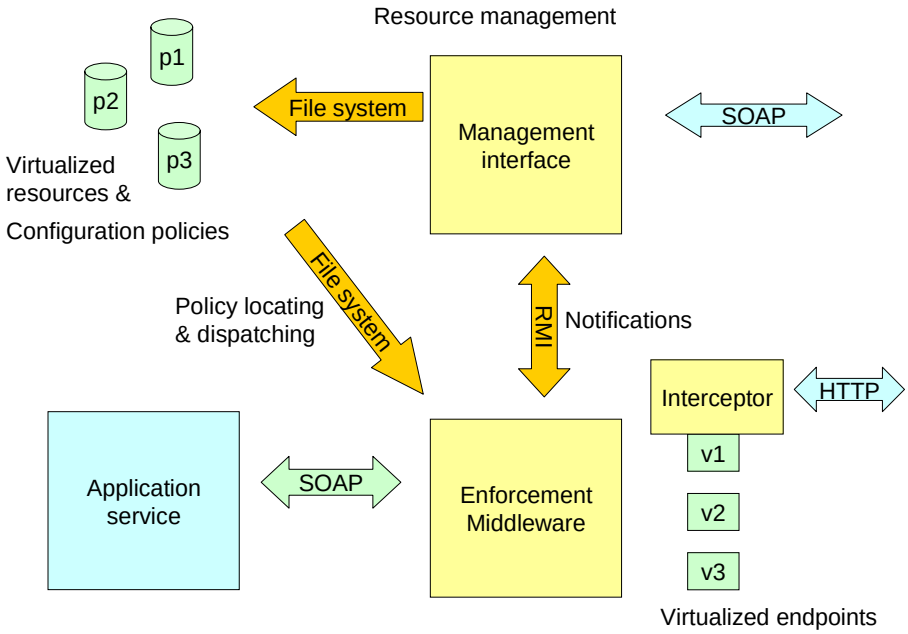


Figure 42 Policy Enforcement Point (PEP): relationship between management, interception and enforcement components

VI.6.b Notification related

Notifications are one major form of communication between participants within a Virtual Organisation and potentially even outside the VO: they encapsulate event-based messages that may be multicasted to a set of parties interested in such kind of information. Since VOs are partially event-driven, in particular with respect to dynamic reaction to SLA violations, reputation changes, environmental conditions etc., notifications mainly contribute to the dynamic management of a Virtual Organisation, even though they may be exploited for any other form of information distribution.

Due to the nature of these messages, it is a recommendation for participants in a VO to host for the respective capabilities, i.e. to send and receive notifications, though this is not a general requirement. Obviously, any service that relies on the respective functionality for information gathering and – implicitly – any service that provides this kind of information, such as Reputation and the VO Policy Service, needs to cater for these capabilities. Notably, the Notification subsystem allows for a very flexible deployment and may be easily outsourced.

The actual subsystem consists of two main components: the Notification Proxy (required) and the Notification Broker (optional). Whilst the former is responsible for the actual reception and distribution of notification messages, the latter's task is twofold: for one, it

acts as a broker for messages with unclear recipients, for message re-routing and/or for reducing band-width. Furthermore, the Notification Broker in TrustCoM acts as a subscription manager that maintains all notification sources and sinks in the VO and updates the Notification Proxies accordingly (see chapter III and Appendix B, section VI.2.d for a detailed description).

1) Information Artefacts

This section details the main artefacts related to notifications and how they may be maintained. We can identify in particular two message types of relevance for the subsystem: those related to notification messages and those related to subscribing to a specific event-type, respectively registering as a publisher of a specific event-type.

Notification

Notification messages carry information about specific events occurring at the sender side. Due to their nature, notifications may carry any form of data as their main content since it strongly depends upon the type of event and its sender. Accordingly, it is recommended that the message content is plain XML without specific restrictions.

Optionally, notification messages may specify the context of the event by which they were triggered – the "topic" or "event name" as they are generally called. Such a topic may be specified in any degree of complexity, ranging from a simple string to a linked hierarchy of its own, depending on the information detail demanded. Note that a higher complexity may reduce the amount of notifications due to more concrete filtering, yet will increase management overhead.

Subscription / Registration

Subscribing an entity to a (potential) producer of specific event-types *directly*, i.e. without an intermediary manager, like e.g. the Notification Broker, implies that the source will send occurring event messages with their occurrence to the subscriber. In the managed case (using a Subscription Manager, or the Broker in the TrustCoM case), the Broker needs to identify all according producers of the event-types specified and forwards the subscription request to each in turn (cf. below).

Accordingly, the subscription data needs to contain information about the recipient's endpoint[26] (URL or EPR), as well as the name of the topic of interest, given that a topic space has been defined.

Registration of notification producers is only of relevance, when an intermediary "broker" takes care of managing the notification dependencies. This information is required in order to identify all sources for specific event with each subscription request. In the direct proxy to proxy subscription case, such a data set is meaningless.

Similar to subscription, the registration data needs to contain the source's endpoint, along with the event name, if defined.

The subscription manager needs to maintain both data sets in order to update all subscription links in case of a change in the structure (e.g. additional producer, removed subscriber etc.)

[26] Note that in the case of using a gateway like interface for message redirection, a handler would be sufficient for this purpose.

2) Information Exchanges

Notification messaging is a generic capability of the TrustCoM framework and as such not restricted to specific lifecycle phases, e.g. a participant may decide at any time during operation to subscribe to a specific event-type as it becomes of interest to his/her business – strictly speaking, this is not considered evolution, as it does not imply reconfiguration of the VO. However, as far as notifications are exploited for VO management purposes (such as monitoring performance etc.), subscription and registration will typically take place with formation (respectively evolution), whilst the actual distribution of messages will occur mostly during the operational phase.

Formation and Evolution

With every member joining or retiring from the VO, the notification relationships need to be updated according to this particulars party's creation, respectively absorption of event type messages. In other words, the new party needs to subscribe to all notification sources of interest and needs to register as a producer of event messages. In the case of managed notification support (Figure 43), the Broker maintains all information about sinks and sources in the VO and updates the subscriptions accordingly. This means that VO Management only needs to provide information about (a) the location of the Notification Broker and (b) the names of the topics the new provider should produce, respectively (c) subscribe to.

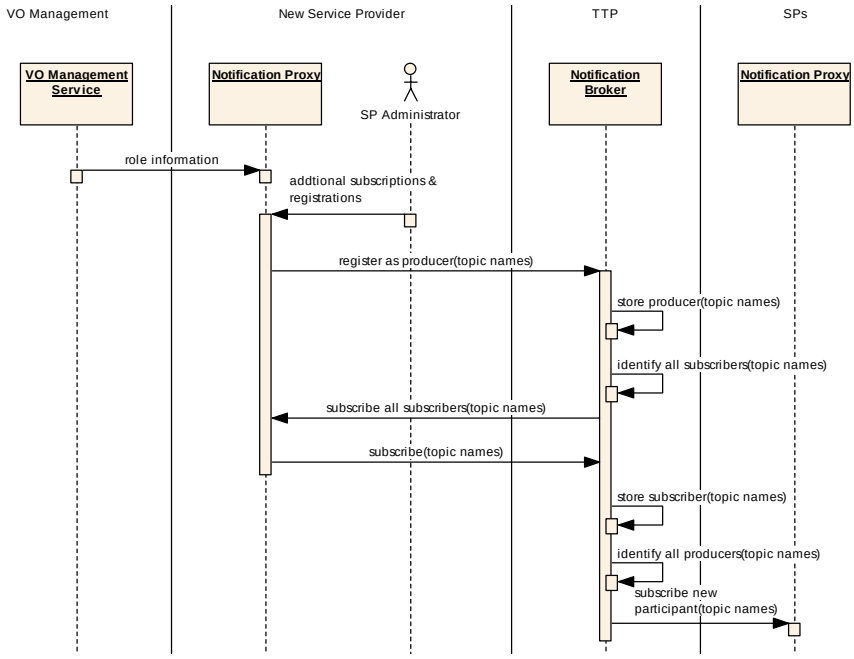


Figure 43: Subscription and registration using a Broker for management support.
Note that all intermediary gateway interactions have been skipped for reasons of simplicity.

In the unmanaged case, the Notification Proxy of the Service Provider needs to be informed about (a) all producers it should subscribe to, including (b) the respective topic names, as well as (c) the names of topics it should produce along with (d) all subscribers that need to be added (Figure 44).

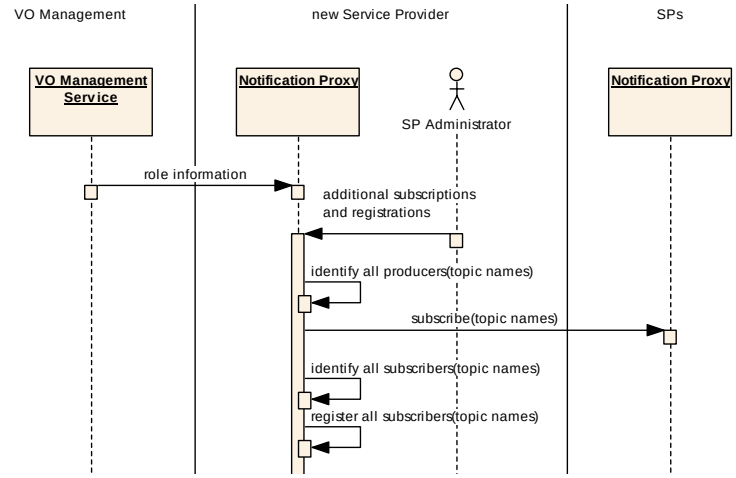


Figure 44: Subscription and registration in the unmanaged case.

VO Operation

For actual distribution of notification messages, the Notification Proxy needs to identify all subscribers of the specified topic to then forward the according event message to each one in turn (Figure 45). As discussed in the previous section, the topic name may be considered optional in a number of cases, e.g. when only one “topic” exists within the VO, when the sender only produces one topic and/or when the consumer knows only one topic.

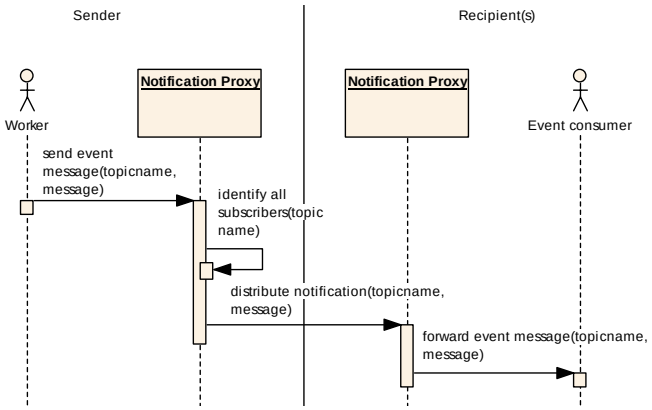


Figure 45: Distribution of notifications without using an intermediary broker.

Within a Virtual Organisation, a Notification Broker may be used to support message distribution, in which case the Broker is registered as (one of) the subscribers to the event source (Figure 46). It is up to the Broker to forward the according message to all (other) recipients. This way, sender and recipient may be hidden from each other (e.g. for confidentiality reasons) and/or the message load of the sender reduced significantly (see also introductory section to this chapter).

Note that any amount of Brokers may participate in the VO and act either in parallel (multicasting the message to multiple Brokers) or even sequential (sending from Broker to Broker).

Great care must be taken in order to avoid duplicating messages at the receiver’s side: two potential approaches may be taken: (a) filtering for duplicate message IDs at the recipient’s Notification Proxy and (b) distinguishing between brokered and unbrokered subscription requests. The first case may result in heavy overloading of the recipients inbound traffic. In

the latter case, the distribution of notifications via the Broker will affect all messages with respect to the according recipient, i.e. neither recipient nor sender can specify which messages to send brokered and which not (at least of the according topic). Another solution may consist in forwarding the list of all (known) recipients to the Broker, so that the latter may identify which endpoints are still valid[27].

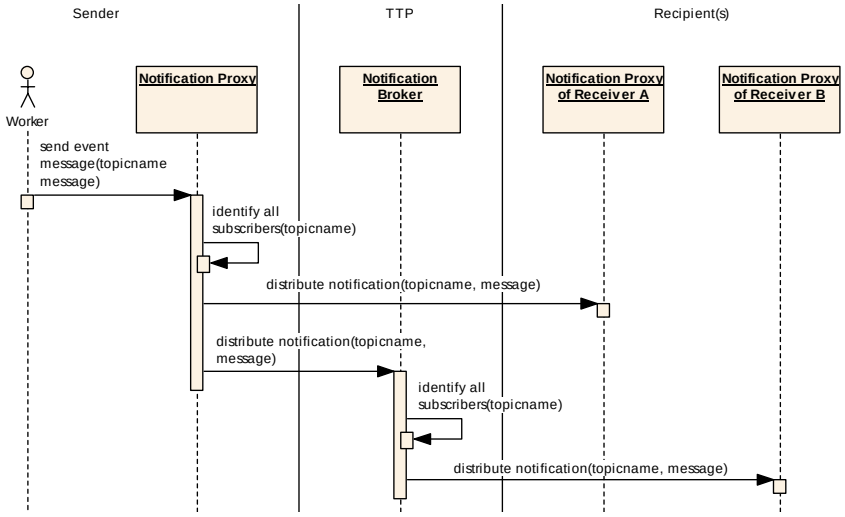


Figure 46: Brokered distribution of notifications.

[27] As this approach is not compliant with the WS-BaseNotification specification, it was not considered in the TrustCoM development efforts.

VII Glossary

Term	Definition
Application Service	*A service that may perform a certain, business oriented task according to a pre-defined workflow.*
Application Service Provider	*Enterprises, companies or individuals that provide Application Services and offer them via searchable registries to customers.*
BP Management System	*The unit responsible for the overall business process that is to be realised by the virtual organisation.*
Business Process	*An abstract workflow that describes the action and tasks a unit has to enact.*
Collaboration Agreement	*→ General VO-Agreement*
Collaboration Definition	*A Collaboration Definition (CD) captures the global view of a business collaboration among roles. It entails roles, high-level activities and interactions.*
Collaboration Definition Template	*A Collaboration Definition Template captures the recurring best practices for a specific well known business collaboration in the format of a CD. It is usually stored in a repository.*
Collaborative Business Process	*A Collaborative Business Process (CBP) entails the set of public and private processes derived from or associated to a CD for each specified role in the CD*
Component	*The smallest functional and/or logical unit. Tightly coupled components interact in order to fulfil a specific task form a → subsystem*
Contract	*A form of convention that designates the behaviour the involved parties commit to. This is principally the legal counterpart to → SLAs.*
Dissolution Phase	*During this → VO lifecycle phase the → Virtual Organisation is dissolved again and all partners are released from their respective bindings (→ contracts, → SLAs etc.)*
Enterprise Network Agreement	*A description of the requirements to be fulfilled in order to become member of an Enterprise Network. This is the basis for the → General VO Agreement*
Evolution Phase	*Sometimes distinguished from the → Operation Phase. During this → VO lifecycle phase, changes to the → Virtual Organisation may occur – this covers in particular the addition & exclusion of individual partners.*
General VO-Agreement	*The "high-level" definition of all the parameters and rules that have to be fulfilled by all participants. This may involve → Contract terms*
Identification	*The → VO lifecycle phase during which potential partners to support*

Phase	*the overall business goal are discovered.*
Formation Phase	*In this → VO lifecycle phase partners are invited to the → Virtual Organisation and they are provided with the necessary information to collaborate. During this phase the VO is actually formed.*
Operation Phase	*The → VO lifecycle phase during which the → Business Process(es) are executed to reach the VO's business goal(s)*
Policy	*Rules defining choices in the behaviour of systems. Within the scope of the TrustCoM project several types of policies are considered.* - *SLA Obligation policies which define the obligations of a party in respect to the provision of a QoS to the other party. These policies trigger notifications when the specified QoS has been violated.* - *Access control policies in the form of authorisation and delegation policies which define who can access services and under which constraints.* *Obligation Policies (in the form of Event-condition-action rules) which define how the VO should adapt to failures, changes in requirements, security events, etc.*
(Authorisation) Policy Decision Point	*Decides which messages are permitted or not depending on the current access control policies.*
(Authorisation) Policy Enforcement Point	*It is the point where the incoming message is intercepted, the tokens provided with the message are verified and an access control decision is requested from the PDP.*
Private Process	*A private process is an executable business process enacted by a BP engine, contributing to the VO's business objective by orchestrating services. A private process is confidential to a VO member domain, the process owner, due to optimisation and associated sensitive information.*
Public Process	*A public process captures the externally visible part of exactly one private process. The public process can be seen as the private process interface with the minimal exposure to let the private process collaborate in a CBP.*
Repository/ Registry	*A database that stores information about (publicly available) services, like e.g. their WSDL, → SLA templates etc.*
Security Token	*Contains authentication relevant information, may also contain access-rights and related data.*
SLA	***S**ervice **L**evel **A**greement: an electronic form of → contract, that is only of limited legal impact. It describes the quality of service that has to be maintained.*
SLA Management System	*Responsible for managing → SLAs – this includes negotiation of the parameters, monitoring & evaluating service performance and*

	enforcing the obligations.
SLA Template	*A document that contains the parameters that can principally be fulfilled by the service that provides the template.*
Subsystem	*A subsystem represents a logical and functional unit of → components that interact in order to fulfil the subsystem's task.*
Supporting Service	*Services that are in themselves not part of the virtual organisation, but that are used by the latter to fulfil certain purposes. Supporting services are mainly → Repositories and Registries so far.*
Trust	*In the sense used here mostly related to "trustworthiness": the expectations put in a service to behave in a particular way. This reflects first of all an evaluation of past performance.*
Trusted Third Party	*Services that participate in a virtual organisation, yet do not directly contribute to the realisation of the overall → Business Process (as opposed to an → Application Service)*
TSC Extension Role	*The TSC Extension Role is part of the BP TSC concept and it configures a TSC task, to perform TSC control functions based on defined subsystem EPRs and parameters.*
TSC Task	*The TSC task is part of the BP TSC concept which provides process control based on the TSC subsystems during process instance runtime.*
Virtual Organisation	*A set of business entities that work together (by message exchange etc.) to reach a common goal – generally represented by an overall → Business Process.*
VO Lifecycle	*The → Virtual Organisation traverses 5 main phases that logically distinguish the actions to be performed. These phases are: → Identification, → Formation, → Operation & → Evolution (sometimes regarded as one phase) and → Dissolution.*
VO Manager	*The central management instance that acts on behalf of the VO-customer. This entity is responsible for "guiding" the VO lifecycle and performing membership-related management tasks.*

VIII Key to diagrams

The following UML diagram types may be used in this document:

- Dynamic modelling by **Activity diagrams** with swimlanes for different sub-systems, or components of subsystems.
- **Component** or **Composite Structure diagrams** to represent the structure of components that form a subsystem (respectively subsystems that form the TrustCoM system in the overview case) and their dependencies to each other.

When diagrams are used they should use symbols and notations defined in the standard Rational Unified Process (RUP); the elements for these diagrams are summarised below.

Summary of Activity Diagram Elements

	Diagram Element	Symbol	Represents
Nodes	Action state	Horizontal capsule	A process
	Decision	Diamond	Next step may be only one of several sucessors
	Swim lane	Parallel vertical lines	A group of related processes
	Synchronization point	Thick bar	All predecessors must terminate before the successor can start
	Object	Object box	An object, component, or subsystem
	Signal receiver	Notched rectangle	The successor cannot be started until the signal is received
	Signal sender	Pointy rectangle	A signal is sent before the successor start
	Initial action state	Filled circle	Predecessor to the first action state
	Final action state	Bull's eye	Final action state
Edges	Control flow	Solid arrow	Pre- and post-decessor relationship
	Message flow	Dashed arrow	Message sent to/from an object
	Signal flow	Dashed arrow	A pair of sender/receiver nodes

Summary of Component Diagram Elements

	Diagram Element	Symbol	Represents

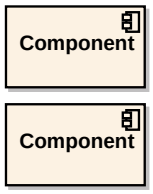
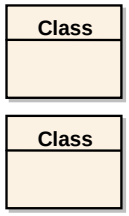
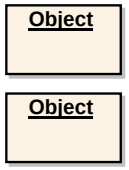
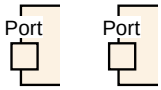
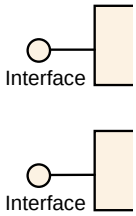
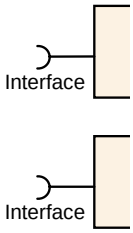
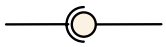
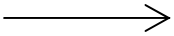
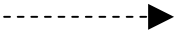
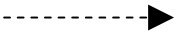
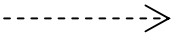
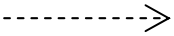
	Diagram Element	Symbol	Represents
Nodes	Component		A modular part of a system, whose behaviour is defined by its interfaces
	Class		Representation of object(s), that reflects their structure and behaviour within the system.
	Interface		A specification of behaviour supported.
	Object		Particular instance of a class.
	Port		A distinct interaction point
	Provided Interface		Interface provided by the component
	Required Interface		Interface required by the component
	Artifact		Physical piece of information used or produced by a system

	Diagram Element	Symbol	Represents
Edges	Assembly		Connection between a provided and a required interface
	Associate		Denotes relationship between two elements.
	Delegate		
	Realise		
	Generalise		
	Dependency		
	Trace		

Summary of Composite Structure Diagram Elements

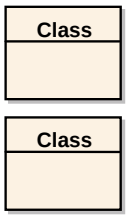
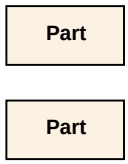
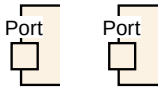
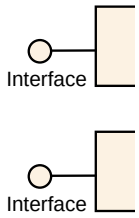
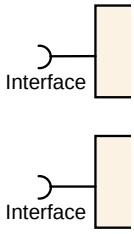
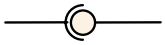
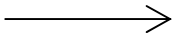
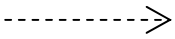
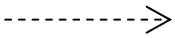
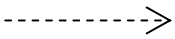
	Diagram Element	Symbol	Represents
Nodes	Class		Representation of object(s), that reflects their structure and behaviour within the system.
	Interface		A specification of behaviour supported.
	Part		Run-time instances of classes or interfaces.
	Port		A distinct interaction point
	Collaboration		

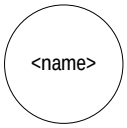
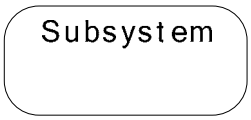
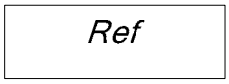
	Diagram Element	Symbol	Represents
	Component provides Interface		Interface provided by the component
	Component uses Interface		Interface required by the component
Edges	Assembly		Connection between a provided and a required interface
	Connector		Communication link.
	Delegate		
	Role Binding		
	Represents		
	Occurence		

This document furthermore makes use of a non-UML based diagram type to depict the so-called “relationship model” introduced with this document. This diagram type reflects the *potential* information transport between components, non-regarding their deployment, respectively actual “usage environment” (see chapter IV for a detailed description of the diagram type).

Though such a model could principally be depicted using UML specific representations, we chose the following symbolic representation to avoid confusion:

Summary of Relationship Model Diagram Elements

	Diagram Element	Symbol	Represents

	Diagram Element	Symbol	Represents
Nodes	Component		Components, parts of the TrustCoM framework – generally services and/or libraries.
	Service		Any (web) service that participates in a Virtual Organisation.
	User		Human beings in a VO (customer, service owner / administrator etc.)
Edges	Message		Passing data
	Trigger		Action invocations (triggers)
	Relationship		Data relationship on subsystem level
Other	Boundary		Logical boundary of a subsystem
	References		Reference to other diagrams

IX References

[1] Preliminary Conceptual Models for the TrustCoM Framework, ID 1.1.2, version 1.0 - http://portal.sema.es/pls/portal30/docs/FOLDER/TRUSTCOM_AREA/WORKFOLDERS/AL1FRAMEWORKDEFINITION/ACTIVITY11CONCEPTUALMODELS/WP1111TRUSTCONTRACTMANAGEMENTMODELS/ID112/ID.1.1.2_V1.0.DOC

[2] TrustCoM Reference Architecture, D09, ID 1.2.4, version 1.0

[3] The TrustCoM Framework V3

[4] Baseline Prototype Infrastructure for the CE Scenario, D10, version 1.0

[5] The TrustCoM Concepetual Models, D16, version 1.0

[6] VO Trust, Security & Contract Management Framework, D18, version 1.0

[7] An Intermediate Assessment of the TrustCoM Framework using the CE Scenario, D20

[8] An Intermediate Assessment of the TrustCoM Framework using the AS Scenario, D21

[9] W3C. Web Services Choreography Description Language, 2005. W3C Latest Working Draft from October 8th, 2005, work in progress

[10] Marlon Dumas and Arthur H. M. ter Hofstede. UML Activity Diagrams as a Workflow Specification Language. In UML '01: Proceedings of the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools, pages 76–90, London, UK, 2001. Springer-Verlag.

[11] Alistair Barros, Marlon Dumas, and Phillipa Oaks. A Critical Overview of theWeb Services Choreography Description Languages (WS-CDL). BPTrends Newsletter, Vol. 3, March 2005

[12] Alistair Barros, Marlon Dumas, and Arthur H.M. ter Hofstede. Service Interaction Patterns: Towards a Reference Framework for Service-Based Business Process Interconnection. http://sky.fit.qut.edu.au/ dumas/ServiceInteractionPatterns.pdf, April 2005.

[13] Roberto Gorrieri, Claudio Guidi, and Roberto Lucchi. Reasoning about interaction patterns in Choreography. In Proceedings of the 2nd International Workshop on Web Services and Formal Methods (WS-FM '05), 2005.

[14] WS-Trust specification from February 2005 (http://msdn.microsoft.com/ws/2005/02/ws-trust/).

[15] SAML 1.1 Assertion specification (http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf).

[16] Web Services Security SAML Token Profile 1.1 (http://www.oasis-open.org/committees/download.php/15256/Web%20Services%20Security%20SAML%20Token%20Profile-11.pdf).

[17] Asit Dan, Heiko Ludwig and Giovanni Pacifici. Web service differentiation with service level agreements. http://www-128.ibm.com/developerworks/webservices/library/ws-slafram/index.html

[18] W. Saabel, T.M. Verduijn, L. Hagdorn and K. Kumar. A Model for Virtual Organisation: A Structure and Process Perspective. Electronic Journal of Organizational Virtualness, Vol. 4, 2002

[19] Bastian Koller and Lutz Schubert (2005),Towards Autonomous SLA Management Using a Proxy-like Approach, *in* R Hirschfeld; R Kowalczyk; A Polze & Mathias Weske, eds.,'Conference Proceedings NODe 2005, GSEM 2005'.

[20] Jochen Haller, Lutz Schubert and Stefan Wesner (2006). Private Business Infrastructures in a VO Environment, *in* Paul Cunningham & Miriam Cunningham, eds., 'Exploiting the Knowledge Economy - Issues, Applications, Case Studies'.

[21] Peer Hasselmeyer, Bastian Koller, Lutz Schubert and Philip Wieder (2006). Towards SLA-Supported Resource Management, *in* 'HPCC06 Conference Proceedings'.