Deliverable

70

WP38

Final Implementation and Validation for the Ad Hoc Dynamic Processes Demonstrator

Paul Kearney, BT

June 2007

Issue 1.0

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

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: AL3	
Activity:	
Work Package:	**WP38**
Task:	

Document title: Final Implementation and Validation for the Ad Hoc Dynamic Processes Demonstrator

Version:	**Issue 1.0**
Document reference:	**Deliverable D70**
Official delivery date:	**May 2007**
Actual publication date:	**8th June 2007**
File name:	
Type of document:	Report
Nature:	Public

Authors:	Paul Kearney, BT (main author)
Reviewers:	David Golby, BAE Systems
	Michael Wilson, STFC
Approved by:	

Version	Date	Description of version and changes
Draft A	22nd May 2007	Outline of deliverable. Limited circulation. Discussed by phone conference 25/5/07.
Draft B	31st May 2007	Substantially complete draft circulated for discussion, comment and additional contributions. Discussed by phone conference 1/6/07. Main missing sections are Executive Summary, Introduction, Evaluation and Conclusions/future work
Draft C	4th June 2007	Changes relative to Draft B: Evaluation and Conclusions sections completed in draft.
Draft D	5th June 2007	Introduction and Executive summary completed. Detail on sequence of events during service invocation added. This is now complete though in need of review and would benefit from additional detail and diagram in places.
Draft E	6th June 2007	Incorporates comments from WP38 team: SAP (Bernhard), ATOS (Ignacio), SICS (Erik)
Draft F	8th June 2007	Revised in response to review by Dave Golby.
Draft G	8th June 2007	Revised in response to review by Michael Wilson. Main changes are additional paragraphs in the conclusions.
Issue 1.0	8th June 2007	As draft G with changes confirmed and comments removed.

Table of Content

Executive Summary

This is the final report covering work on Work Package 38 of the TrustCoM project and is the companion to the Ad Hoc Dynamic Processes Demonstrator. The main focus of the demonstrator is the Virtual Hosting Environment (VHE) business concept and platform. Used here, the term VHE describes a platform operated by a hosting service provider that implements the kernel of the TrustCoM Framework. This provides a hub for communities of enterprises (enterprise networks, ENs) cooperating via software services. In particular, we are interested in ENs where cooperation is in terms of creating composite software services combining elements contributed by different EN members. In this context, a Virtual Organisation (VO) is one of the composite services plus the service providers involved and the relationships between them. The VHE provides a secure and trustworthy environment for assembling and operating such VOs. It is an application-neutral platform that can be specialised to the needs of ENs operating in particular market sectors. The VHE operator acts as an outsourcer supplying B2B collaboration infrastructure to the EN. The EN organiser and members are then able to focus on business and application issues.

We expect that the Virtual Hosting Environment concept will be widely taken up. It offers substantial business opportunities to service providers, especially existing operators of telecommunications networks, data centres and application hosting facilities, and very much in line with forward-looking business models of a number of TrustCoM partners. The existence of VHE implementations will create opportunities for companies and other organisations to form ENs on a commercial or public service basis. Without a VHE an EN would need to create a bespoke infrastructure and management software or integrate its members' resources in an ad hoc fashion. Both of these options require time, specialist expertise, and considerable investment. In turn, the availability of these safe environments for co-operation will remove barriers to the blossoming of an ecosystem of innovative small companies and be a considerable stimulus to European economic prosperity.

The main objective of the demonstrator is to make progress towards commercial exploitation of TrustCoM framework by means of a prototype VHE that is suitable for explaining the concept and stimulating a dialogue with parties interested in business opportunities as well those wanting details of the technical solution. To provide an application context for the demonstrator, two application scenarios have been studied. They concern ENs in the eLearning and on line music retail sectors.

Despite a highly compressed timescale due to delays in other TrustCoM work packages, we have successfully implemented the planned core functionality. It has not been possible to give any demonstrations and hence to receive feedback from interested parties outside the project. Consequently, the evaluation given in the report is a subjective one by the team involved in producing it. The primary objectives have been achieved. Experience during development of the demonstrator has reinforced our view that the VHE concept and business model are very relevant to emerging business requirements and well-aligned with the business strategies of TrustCoM partners. Furthermore, availability of production implementations of the VHE will open up opportunities for innovative small businesses to experiment with new business models, products and services, which is a key objective of the FP6 programme.

The demonstrator definitely qualifies as a useful early prototype for a VHE service platform. It implements a set of functions that are core to the role of the VHE, and the modular architecture allows for easy extension of functionality through addition and upgrade of services. Furthermore, the core functions represent a significant advance over the state of the in secure exposure and rapid integration of software services in an inter-enterprise environment. While much work would still be required to turn the demonstrator into a production service platform, the key features are in place and have been shown to function as advertised. The evaluation section includes an indication of further development required.

Beyond the end of TrustCoM, the demonstrator will be used by the 'industrial' partners in TrustCoM to gain commercial benefit from the project's results. It will be used widely to stimulate a dialogue with business representatives over novel product and service propositions. There is also always a demand for demonstration of advanced technology to the customer community to provide thought leadership, and also to elicit feedback that will help define product roadmaps. Activities such as this are a necessary pre-cursor to commitment of the investment required for development a prototype production platform. Even if such investment does not materialise the insight gained is highly valuable if it helps shape strategic direction and product and service roadmaps.

1 Introduction

This is the final report covering work on Work Package 38 of the TrustCoM project and is the companion to the Ad Hoc Dynamic Processes Demonstrator.

The main focus of the demonstrator is the Virtual Hosting Environment (VHE) business concept and platform. Used here, the term VHE describes a platform operated by a hosting service provider that implements the kernel of the TrustCoM Framework. This provides a hub for communities of enterprises (enterprise networks, ENs) cooperating via software services. In particular, we are interested in ENs where cooperation is in terms of creating composite software services combining elements contributed by different EN members. In this context, a Virtual Organisation (VO) is one of the composite services plus the service providers involved and the relationships between them. The VHE provides a secure and trustworthy environment for assembling and operating such VOs. It is an application-neutral platform that can be specialised to the needs of ENs operating in particular market sectors. The VHE operator acts as an outsourcer supplying B2B collaboration infrastructure to the EN. The EN organiser and members are then able to focus on business and application issues.

We expect that the Virtual Hosting Environment concept will be widely taken up. It offers substantial business opportunities to service providers, especially existing operators of telecommunications networks, data centres and application hosting facilities. The existence of VHE implementations will create opportunities for companies and other organisations to form ENs on a commercial or public service basis. Without a VHE an EN would need to create a bespoke infrastructure and management software or integrate its members' resources in an ad hoc fashion. Both of these options require time, specialist expertise, and considerable investment. In turn, the availability of these safe environments for co-operation will remove barriers to the blossoming of an ecosystem of innovative small companies and be a considerable stimulus to European economic prosperity.

The VHE idea is very much in line with forward-looking business models of a number of TrustCoM partners. For example, BT plans to offer a range of added value business services and collaboration platforms on top of its next generation 21st Century Network (21CN). An end-to-end IP-based network, 21CN will consolidate BT's complex network and systems. However, 21CN is more than a network transformation; it is a radical overhaul of products, systems, process and a fundamental remaking of BT's business. The VHE fits in very well with this strategy for the future Digital Networked Economy.

The main objective of the demonstrator is to make progress towards commercial exploitation of TrustCoM framework by means of a prototype VHE. To serve as a demonstrator, the prototype must be suitable for explaining the concept and stimulating a dialogue with parties interested in business opportunities related to the VHE concept as well as details of the technical solution. To provide an application context for the demonstrator, two application scenarios have been studied. They concern ENs in the eLearning and on line music retail sectors.

The report is organised as follows. First the business motivation for the VHE concept is discussed in more detail. This includes an outline of the types of EN the VHE is intended to support. The VHE platform architecture is then described. This architecture is one of many configurations in which the TrustCoM Framework can be instantiated. Next a section on the implementation of the demonstrator covers (amongst other things) how aspects of the TrustCoM Reference Implementation have been developed, adapted and integrated to create the demonstrator. The demonstrator functionality is then described in terms of use cases corresponding to the main lifecycle stages of an EN. This section also includes a description of the application scenarios. This is followed by an evaluation of the demonstrator including directions for future development. The concluding section includes an indication of how the demonstrator will be used as a vehicle for commercial exploitation beyond the end of TrustCoM.

2 Business motivation

The grand vision behind service orientation has long been one of rapid composition of complex and sophisticated distributed applications by connecting functional capabilities deployed as software services. The term "*software services*" is used here to denote a functional capability that is deployed and made accessible to other software services. In contrast to re-use of software components from libraries or packages, a single deployed software service is used in principle in many distributed applications (effectively composite services) simultaneously. While this idea has value even within a single large enterprise, the real paradigm shift requires software services provided by different enterprises to be available to be combined to within larger distributed services or applications. The simplistic image of consumers of services searching public directories to identify services matching their requirements published by any enterprise in the world, and these integrating seamlessly into their applications is fraught with difficulties including the following:

- Adherence to standards does not guarantee technical interoperability. Profiles are required to document how the standards are applied in combination in particular usage scenarios. These profiles often need to be specific to classes of application or market sector.
- Some form of service ontology is required to enable compatible services to be matched. Until work matures in the semantic web community, for example, service ontologies must be agreed with well defined communities of interest.
- Security and access control. There is a distinct possibility that a mis-match of access controls, identity infrastructures, etc. will mean that either things do not work, or else gaps appear that allow access to services by unauthorised parties and other security breaches.
- Trust. An open community always contains untrustworthy parties. Well behaved entities need to be protected through mechanisms helping them to distinguish between trustworthy and untrustworthy partners, and ones that prevent or discourage untrustworthy behaviour.
- Business considerations. A cross-enterprise service-oriented application / composite service is a business collaboration as well a distributed software system. This poses requirements for management of contracts and SLAs, accounting for the purposes of billing and revenue sharing, compliance, assurance and auditability, etc.

The above arguments dictate that for practical purposes, virtual organisations (VOs) are formed within well-defined communities of interest rather than in fully-open communities. Note that in the context of this report, a VO is best thought of as a two-layer entity. The lower layer is a composite service oriented application implementing a business process, and the upper layer is a composite business entity made up of the providers of the services used and the agreements and relationships connecting them. The term used within TrustCoM for a community of interest in which VOs can form is 'enterprise network' (EN).

Two contrasting types of EN are considered in TrustCoM. The first is an extended supply chain or industry market place in which a dominant customer (e.g. a systems integrator or retailer) or group of customers can dictate rules and conventions. This is considered primarily by the Collaborative Engineering demonstrator in Work Package 37. The second is more in line with 'new wave', 'Generation 2.0[1]' business models in which enterprises of any size can offer software services within a community of peers.

We envisage the emergence of new forms of business collaboration that could not exist without the new generation of open standards-based technology. In particular, we are interested in loosely-coupled, geographically dispersed communities of individual knowledge workers and innovative small businesses. Many believe that innovative small businesses will be the engine for future economic growth in Europe. However, individually such businesses lack the scale and breadth to compete with large enterprises. Visionary scenarios have been proposed (e.g. [1][2][3]) in which small business / individuals with

[1] A socio-technical phenomenon that has been given the label Web2.0 has emerged since 2004. According to Wikipedia, Web2.0 'refers to a perceived or proposed second generation of Web-based services—such as social networking sites, wikis, communication tools, and folksonomies—that emphasize online collaboration and sharing among users'.

complementary capabilities are able to cooperate flexibly, supported by appropriate new technology, to provide a range of services currently only within the scope of large corporations. Indeed businesses have founded based on this type of model (e.g. Elance [4], guru.com [5], hotdispatch.com [6]]).

Again we envisage an EN that is focussed on a market sector so that an ecosystem of compatible and complementary services can be built up. In this case it is more likely focused on offering attractive new services to end-users rather than offering efficiencies to a dominant customer. A number of roles can be identified in this software service marketplace, e.g.

- An entity that runs the service marketplace that is the EN. It sets the terms of membership and rules of operation. We will refer to this role later as the ENFounder. This is person or organisation is likely to have sector specific expertise.
- Provider of basic capabilities. We will use the term 'capability' to refer to a chunk of re-usable functionality that is or could be provided as a software service. A 'basic' capability is an individual 'lego brick' out of which more complex capabilities/services can be constructed. Basic capabilities tend to be general purpose / application neutral
- Provider of aggregated services. An aggregated service combines a number of services implementing basic capabilities with additional value-adding functionality to create a higher-level capability. It is likely that an aggregated service will have a more specific purpose than a basic capability, and hence be usable in fewer circumstances. However its immediate value is higher in circumstances where it can be applied. Multiple levels of aggregation are possible.
- Portal provider. This role specialises in interacting with end users and helping them explore their requirements and find / create services that fulfil them
- Infrastructure provider. Inevitably this role has to be played by a substantial enterprise as considerable investment and specialist technical expertise is likely to be involved. The investment may need to be recouped by supporting many ENs over the same infrastructure. Separating out the role of infrastructure provider from the other roles is an important factor in making them accessible to enterprises of all scales.

It is worth noting that aggregated services can be created relatively rapidly by connecting together ('choreographing') pre-existing services provided these have been designed well for re-use. Indeed it might be possible for technically unskilled people to construct aggregated services given the right tools. Since little cost and time is involved, it is not necessary to generate a high volume of business to make a profit, and similarly, the penalty of failure is low. Service providers can therefore afford to speculate and experiment to find niche markets. We can therefore expect a much wider variety of services being offered by a broader range of providers. This expected flattening and broadening of the service spectrum is referred to as 'The Long Tail'.

In Web2.0 circles, an aggregated service is known as a 'mashup'. Mashups tend to use simpler web service technologies and standards than the WS-* and SOAP-based ones on which TrustCoM is based. They also are more user-interface focused and used to create collaborative applications that support social interaction among their users, and are often also modifiable by their users. Nevertheless, conceptually, the mashup is very similar to the aggregated service that the WP38 VHE is designed to support, and the popularity of Web2.0 supports our belief in the emergence of a new business model based on agile collaborations of small enterprises. The type of application envisaged for the VHE is somewhere between the current use of web services to support relatively static B2B business processes and the fun Web2.0 applications where ease of use and customisation is paramount. The VHE is aimed at the middle ground of serious business application of dynamically aggregated services in a secure and trusted environment.

Two application scenarios have been used as examples in WP38. They deal with important classes of 'long tail' aggregated service: highly individualised services and on-line retail environments serving specialised communities. The first of these is set in the eLearning sector, and the EN is made up of small companies providing services in that area. The basic service providers supply course modules and other services that can be combined to create a customised on-line learning experience tailored to the needs of an individual user. The aggregated service in this case is the customised training course, and the VO consists of the various companies involved in delivering it. Construction of the course is driven by interaction between the user and a specialist service that gathers requirements, proposes a course structure and selects and integrates the constituent basic services. The course delivery could be managed by the same entity or another service provider.

The second example is set in the online retail music sector. The aggregated services are virtual music stores serving specialised markets or communities of interest. We can imagine such stores growing up around cult-status bloggers or on-line communities e.g. in MySpace. The basic service providers include copyright owners of musical recordings or their representatives who make these recordings available online, and syndicated blogs or review sites. The virtual music stores reach agreement with music providers enabling them to act as re-sellers of bundles of recordings from their catalogues. The relationship should be of mutual benefit. The virtual music stores will provide large copyright owners with outlets for more obscure items in their catalogues that are of interest to relatively small numbers of customers. Small publishers, including artists publishing their own work, will no longer need to rely on their own web presence. Music connoisseurs and enthusiasts will have on-line forums where they read independent reviews, share views with like-minded souls and be able to purchase new tracks they are recommended. The community served by a virtual music store will value its independence of the major copyright owners, such that any attempt by them to reach exclusive deals or influence editorial content will be counterproductive.

The main focus of the demonstrator is not on specific enterprise networks, aggregated services or market sectors, but rather on the Virtual Hosting Environment (VHE), a generic, network-based platform that can be used to host enterprise networks. We envisage this being operated by a large enterprise with an existing investment and market presence in provision of network-based infrastructure services, for example a network and ICT service provider such as BT. An operational VHE would give a service wrap to a package integrating communication infrastructure, service oriented-middleware, commonly-used generic basic services, and mechanisms and tools to support the setting up, management and operation of the EN. The direct customer for VHE service would be the enterprise or consortium founding and running the EN. The availability of a network-based VHE lowers the barriers to launching an EN and puts the opportunity within the reach of small enterprises and individuals.

Requirements on the VHE include:

- Ease of management of the EN and its infrastructure by an EN operator that does not have technical expertise
- It must be quick and easy to set up new VOs
- There must be a balance of control between enforcement/detection of non-fulfilment of obligations and autonomy of the partners
- The VHE must support multiple simultaneous VO with overlapping membership, which means access control and other mechanisms must be flexible and context dependent

The VHE operator can earn revenue from ‘rent’ received from the EN and usage-based charges for facilities, including its own basic services. However, there are also likely to be synergies with its larger business that may lead it to subsidise the EN in some way. Many large companies are currently looking to harness the innovative capabilities of small enterprises and individuals by encouraging them to create novel applications incorporating basic services they provide. Examples include Google [7], the BBC [8] and Amazon [9].

In the particular case of the VHE operator being a network service provider, the VHE is envisaged as a facility that is integrated with a next-generation, converged, multi-service network (NGN) such as BT’s 21st Century Network (21CN). The VHE provides a way for the NGN operator to offer higher level, added-value services to its business customers. Conversely, the VHE can make use of the NGN service-oriented capabilities (e.g. authentication, billing, etc.) as well as the ability to integrate communication services with application services, and use of the network to send web service messages.

A recent development that will make investigation of integration with the 21CN capabilities easier, is the availability of the BT Web21C Software Development Kit (SDK) Beta release [10], which has been development in conjunction with Microsoft [11, 12]. Web21C is a project to make 21CN capabilities easily available to e.g. third party developers of web service applications. In Phase I, BT has already exposed a number of capabilities to developers, and a lot more functionality will be made available as time progresses. The current SDK beta release is for Visual Studio and is a set of libraries and controls for .NET that makes it simple for developers to consume Web Services exposed by BT. The Web21C SDK abstracts the services interface and serialization classes by providing the developer with a simple object model to interact with.

Controls allow the developer to easily add functionality to both web and windows forms by dragging the control onto a surface and providing basic configuration options. The Web21C SDK implements features of WS-Security and WS-Trust

The following functionalities are accessible in the current version:

- Short Message Service - The Short Message Service (SMS) allows the application developer the ability for individuals to send SMS messages.
- Voice Call - The Voice Call service allows application developers to add the ability for individuals to place phone calls from their application.
- Conference Call - The Conference Call service allows the application developer the ability for individuals to place and control conference calls.
- Presence - The Presence service allows the application developer the ability to store and retrieve an individual's current status and availability for communication.
- Authentication - The Authentication service allows the application developer to create and control an authentication realm for their application. This includes management and authentication of users.
- Information About Me - The Information About Me (IAM) service allows the application developer a way to store and retrieve data about an individual in key value pairs.
- Location - The Location service allow the application developer to add the ability to determine the geographic location (latitude, longitude, altitude) of a mobile device. Currently the Location service only operates in mainland UK with BT issued mobiles, but with service providers partnering all the time, the location service will very soon increase.

The following section describes the prototype VHE developed within TrustCoM. Its main added value relative to other service aggregation platforms lie in its ability to deal with virtual organisations as overlapping business collaborations between autonomous enterprises rather than just providing facilities for integrating loosely-coupled services.

3 The Virtual Hosting Environment

The previous section introduced the VHE as a platform for hosting enterprise networks and the VOs that form within them. Here we look at the concept in more detail and discuss how the elements of the TrustCoM Framework map onto it. While the VHE concept and architecture draw heavily on the TrustCoM framework and Reference Implementation, it is important to remember that the WP38 demonstrator is not simply a development of the corresponding AL2 testbed. Its purpose is not to illustrate as many features of the Framework and Reference as possible, but to demonstrate a commercial concept (the VHE) enabled by the TrustCoM framework and to make progress towards a pilot implementation of the concept usable in business trials. The VHE architecture presented here, is a specialisation of the Framework – specific choices are made within design space defined by the Framework. Much of the section is a summary of material covered previously in [15], and the reader is referred to that document for a more detailed discussion and justification of decisions.

The section begins with a high level description of the VHE abstract architecture. This is followed by a summary of the core functionality required for VHE for the purposes of prioritising features to be implemented in the demonstrator. The remainder of the section looks in more detail at key elements of the VHE architecture and how they are used to deliver the functionality.

Discussion in this section remains at a relatively abstract level. A more concrete description of the demonstrator implementation is including in the subsequent section.

3.1 Abstract VHE architecture

The VHE is essentially an 'unpopulated'[2] instantiation of the generic aspects of the TrustCoM framework packaged as a hosted service. The relationship view of the architecture specified in TrustCoM Framework [13] defines the following logical categories of service in addition to the application services: VO Management, BP Enactment, SLA Management. Policy Services, Trust and Security Services, and EN/VO[3] infrastructure. The framework permits many variations regarding:

- What services within these categories are instantiated and in what form;
- How the service instances are deployed over partner and third party sites. Since the default mode of communication among these services is via SOAP messages, many options are available here.

Example implementations of these services were produced during an earlier phase of TrustCoM and are being made available under open source licenses as the TrustCoM Reference Implementation [14].

The abstract deployment architecture for the VHE features a clear separation between application services contributed by the members of the EN and the generic capabilities of the Framework embodied in the VHE, and greater integration of the services within the VHE. We consider each of the EN members to operate one or more 'sites' (for simplicity, exactly one unless otherwise specified) at which their application services are deployed. In reality, the EN members could use a third party to host their site, and this third party may also be the VHE operator. This does not alter the key point that the application services are outside the VHE.

We consider each of the partner sites to be linked and protected by one or more TrustCoM gateways. These host TrustCoM services provided on a per-partner basis integrated via TrustCoM gateway infrastructure. They perform a number of functions including:

- credential conversion
- enforcement of access control and other policies
- monitoring / event logging and notification

[2] i.e. it needs application services, meta-data, etc. to complete and customise it.

[3] 'the EN/VO' is used here as a short-hand for 'the EN or VOs formed within it as appropriate'.

- encapsulation and virtualisation of services

In the VHE architecture, the gateway is an important modular deployment unit.

As is common for entities at the boundary between network infrastructure and customer sites, the gateways could in principle sit within the network, be part of the customer's own facilities but conform to standards accepted by the network, or be 'customer premises equipment' owned by the network but located on the customer site. In the case of the current demonstrator, we consider the normal case to be that they are owned and operated by the VHE operator, but with certain management rights delegated to EN members. This configuration locates all TrustCoM specific aspects of the architecture under the ownership of the VHE operator (for whom it is a re-usable asset) and hence minimises capital investment, disruption for the partners and also time to set up a EN.

In addition to the (per-partner) gateways are a number of services provided at the EN and/or VO level. These include central membership, service and VO management services. Together with the gateways, they constitute the VHE proper. Each gateway is relatively closely coupled internally, as is the central EN/VO Management block. The units are loosely coupled to each other, however. This means there is still considerable flexibility in deployment options for the VHE operator, but details of the deployment can easily be hidden from the EN partners and founder.

This overall situation is shown in Figure 1. The distinction between the two VHE operator-hosted EN Central Management sites will be clarified later. The sites with names beginning 'GW' are gateways. The gateways labelled GWVM and GWTTP (VM for VO Management and TTP for Trusted Third Party) protect access to the central services. They may be omitted in some variants because the sites they are associated with have a more trusted status. Consequently they are shown faded and with dashed borders. Figure 2 shows an alternative view that emphasises that the VHE acts as logical hub for the EN, though in fact its deployment is likely to be distributed. The network of gateways forms the foundation of the VHE platform and holds the key to providing its core functions. It enables those web services that the EN members have made available to communicate with each other under highly controlled and VO context specific circumstances. It is essentially a SOAP message bus that is highly configurable and into which a variety of enforcement and monitoring elements can be deployed. These elements act on messages passing across the infrastructure performing operations such as routing, verification and validation, transformation (e.g. credential chaining), and monitoring / logging / reporting. The order of operations and detailed behaviour of the elements is under the control of rule-based policies that may themselves be modified in response to observed behaviour.

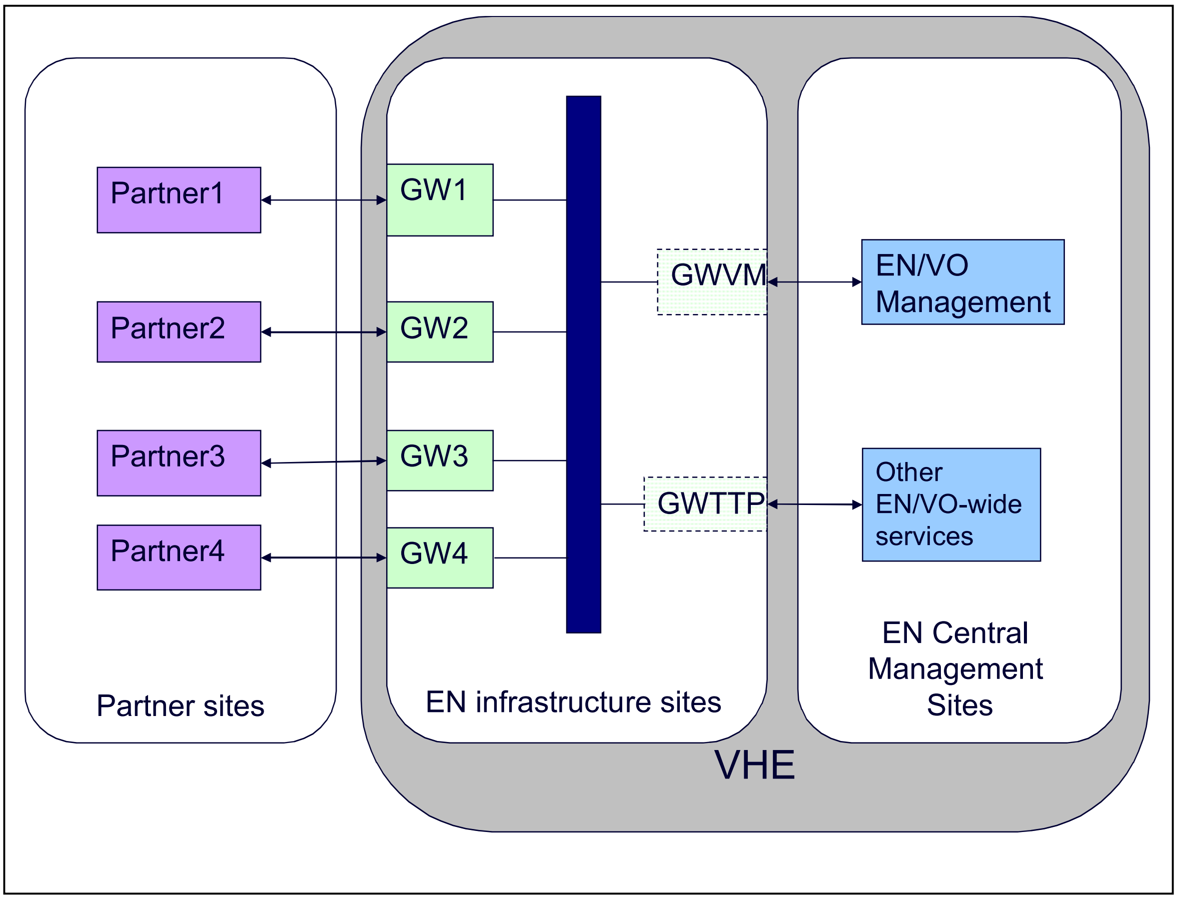

Figure 1: Abstract demonstrator architecture

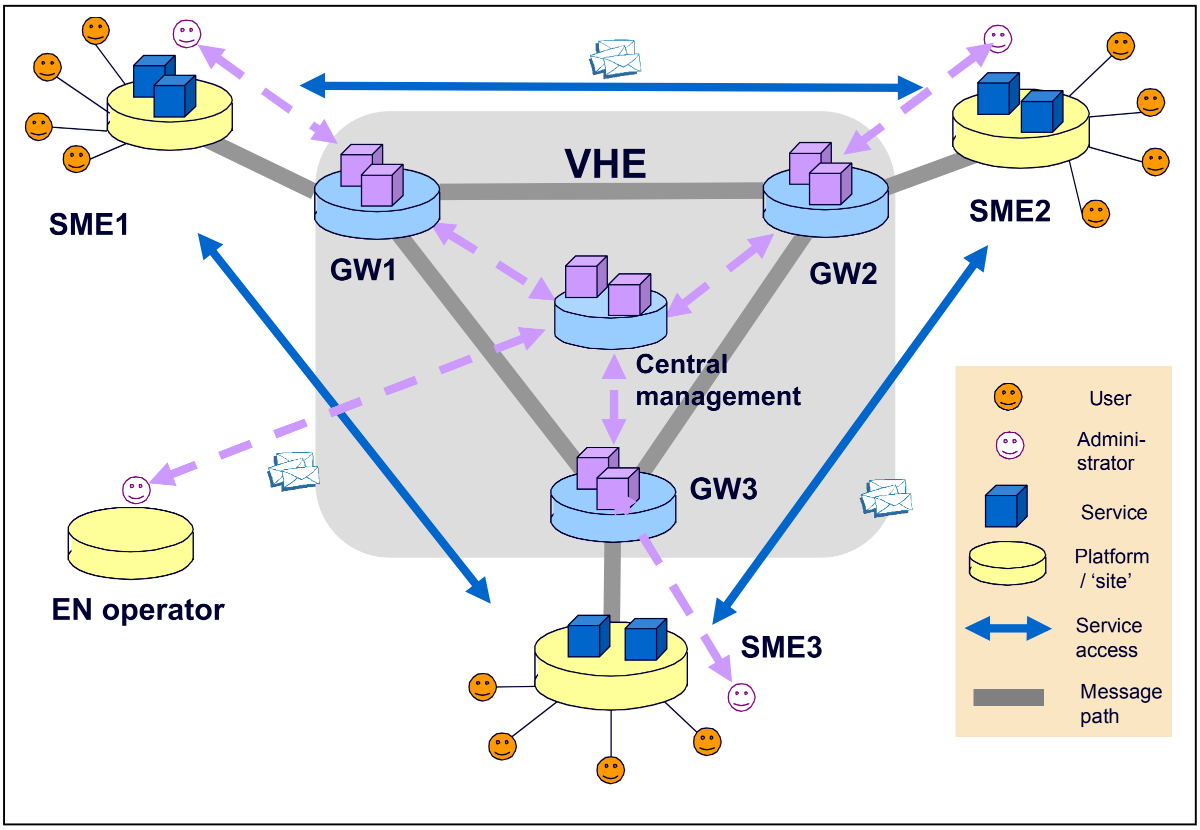

Figure 2: Abstract demonstrator architecture – alternative view

To gain a more detailed appreciation of the architecture and the way the TrustCoM framework functionality maps onto it, it is useful to follow the practice adopted in telecommunications network architectures and distinguish between data, control and management 'planes'. The data plane is concerned with actually delivering the services. It contains the providing and consuming application services and the data paths that connect them, typically via intermediate nodes. The control plane is concerned with controlling the way the services are delivered and ensuring that they are delivered. It contains signalling data flows between local control elements associated with the intermediate nodes. The management plane is concerned with overseeing the operation of the network and communicating policy and configuration changes to the control plane entities.

Applying this model to the TrustCoM architecture, the data plane consists of the application services and policy enforcement points (see Figure 3). The control plane (see Figure 4) contains token services, policy decision points, and analogous services in the SLA Management and BP Enactment sub-systems, for brevity, these are referred to as policy decision services in the figure. The management plane contains EN/VO Management-related services (see Figure 5). In each of these three planes we can distinguish between elements that are under the control of EN/VO members, those that are under the control of the EN/VO or its management, and those that sit on the boundaries between the zones of influence of the partners and the EN/VO (see Figure 6).

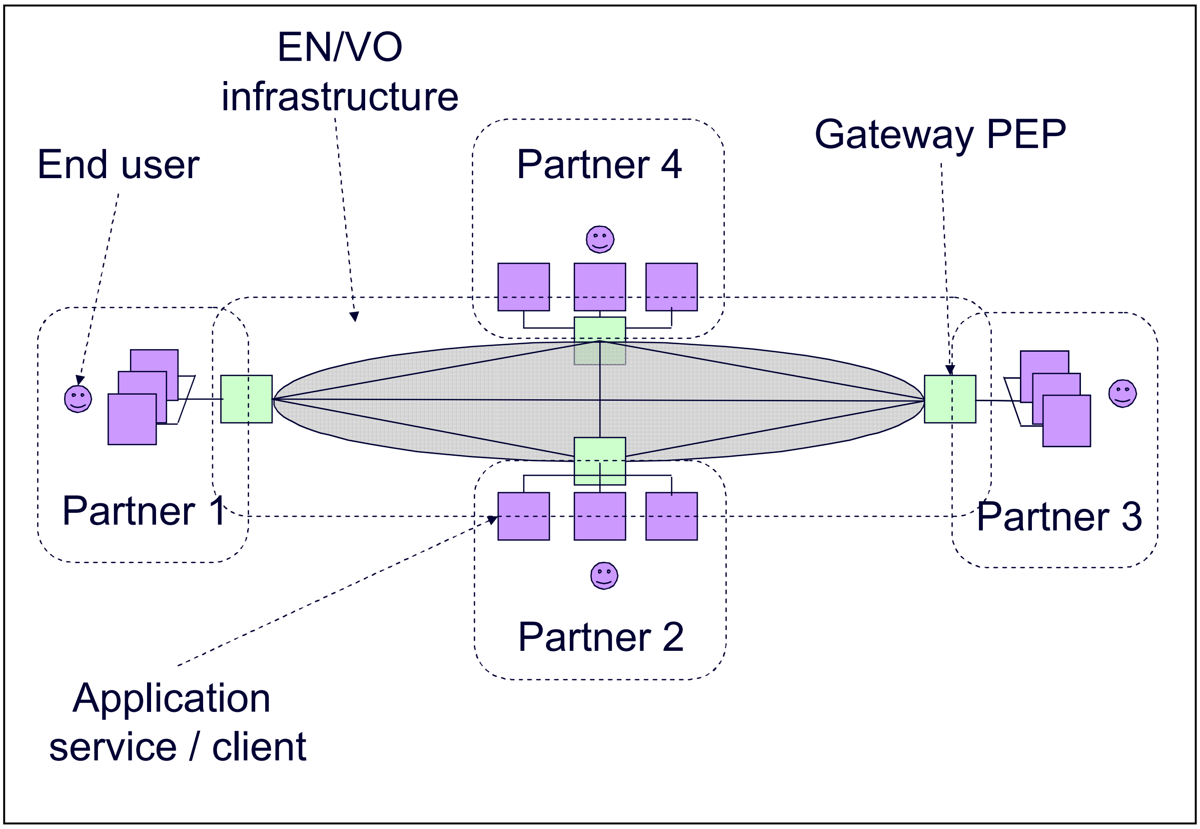

Figure 3: TrustCoM data plane

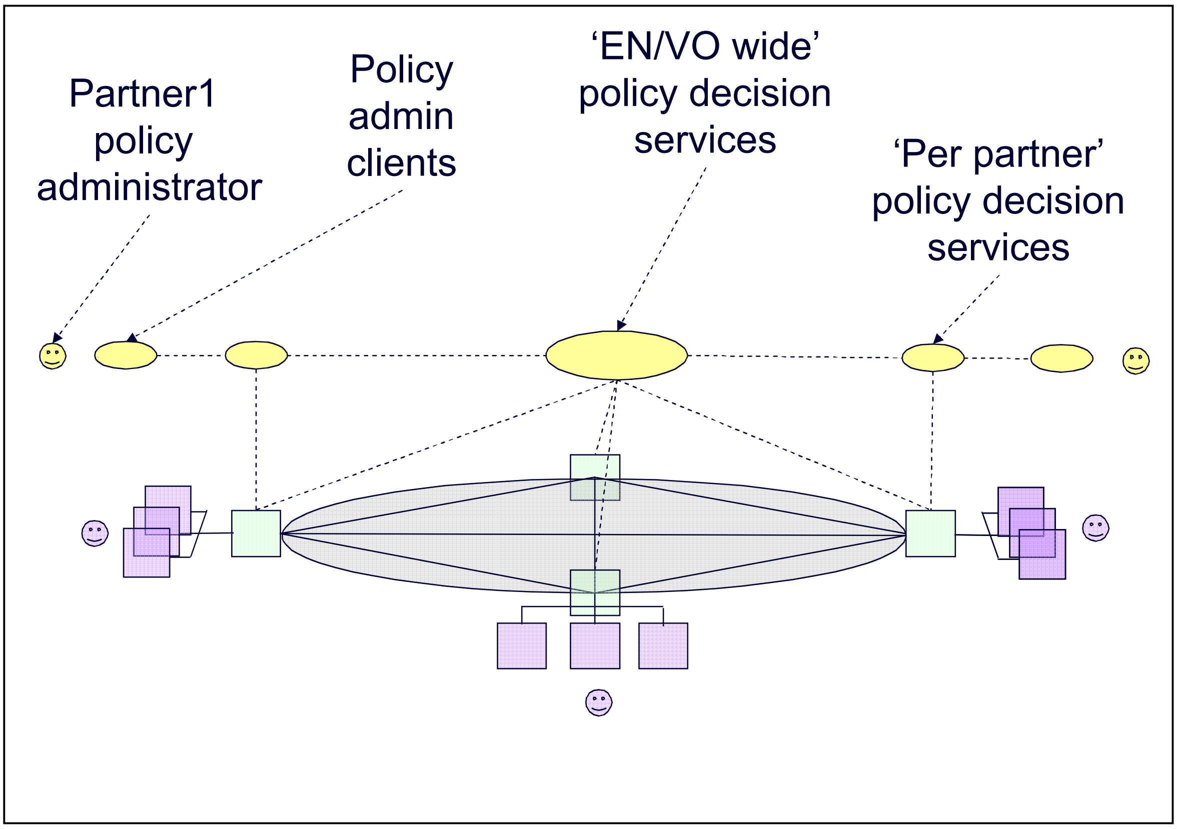
Partner1 policy administrator
Policy admin clients
'EN/VO wide' policy decision services
'Per partner' policy decision services

Figure 4: Adding the TrustCoM control plane

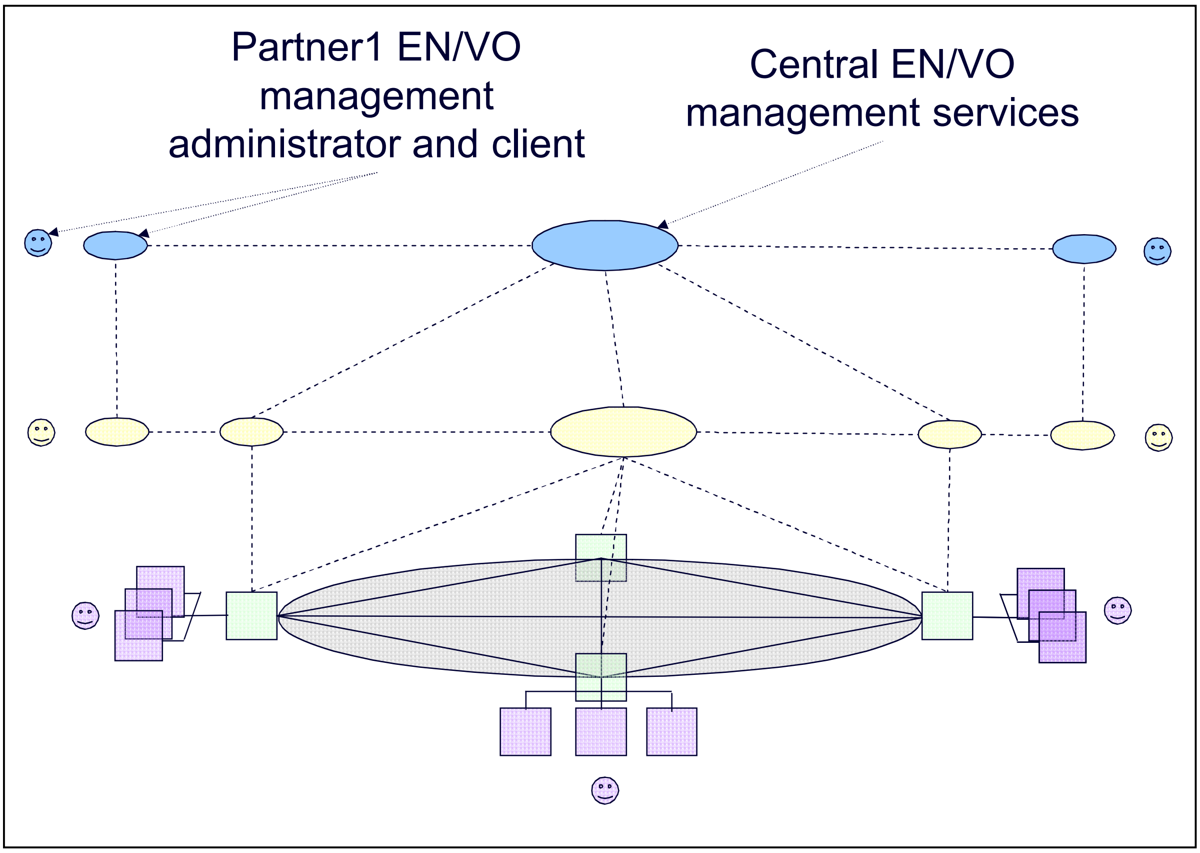
Partner1 EN/VO management administrator and client
Central EN/VO management services

Figure 5: ... and then the TrustCoM management plane

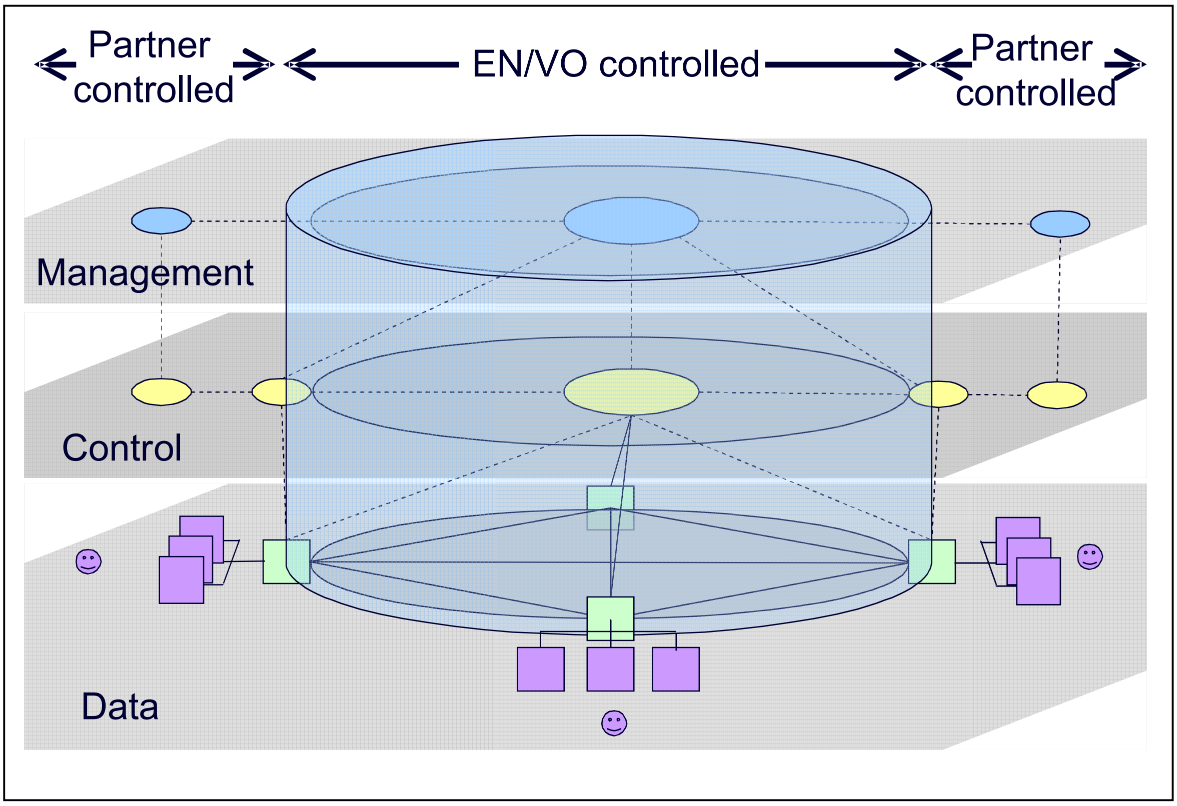

Figure 6: The stacked TrustCoM planes

3.2 Core VHE functionality

Not all the functionality available in the TrustCoM reference implementation is included in the demonstrator as planned for completion with TrustCoM. This is because one the main design objectives was to create a well-integrated, rounded basic platform containing the essential facilities required, and to which other services could easily be added. Thus priority was given to fundamental features that are useful in a wide variety of contexts (the 80-20 rule), and indeed can be built upon to create some of the 'missing' functionality. The three fundamental blocks of functionality to be implemented are Fundamental Access Control, Fundamental Monitoring and Accounting, and Basic EN Membership and Central Service Management. They are described and the choices justified in D38 [15]. The dependencies and priorities accorded to these are shown below in Figure 7. Also shown is an indication of a first generation of extensions that would have been implemented within TrustCoM had time allowed.

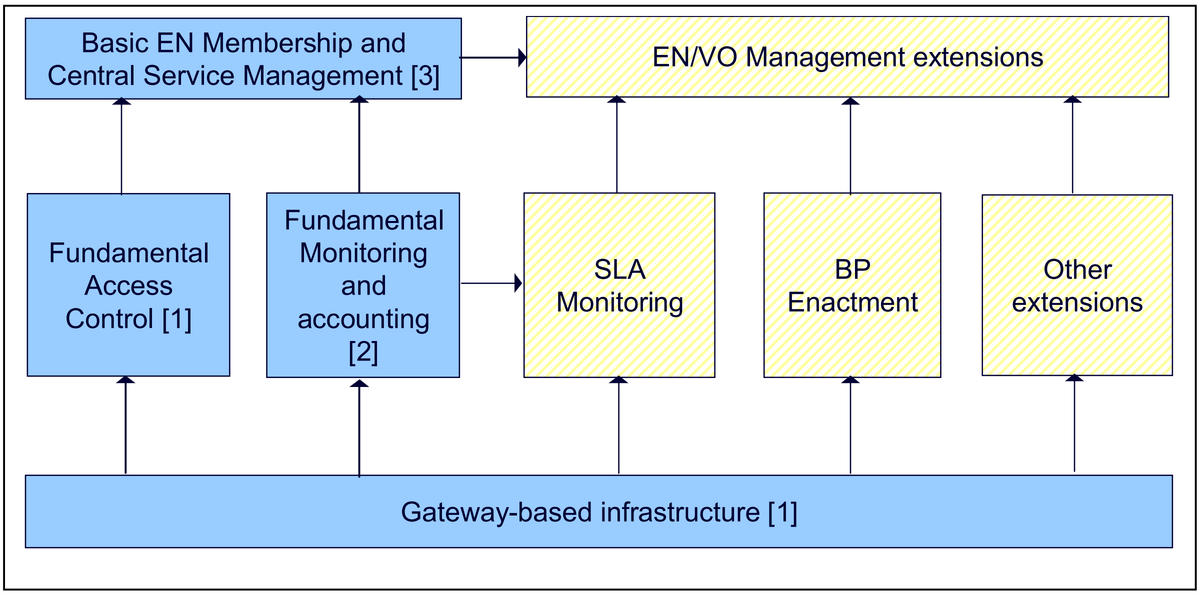
Basic EN Membership and Central Service Management [3]
EN/VO Management extensions
Fundamental Access Control [1]
Fundamental Monitoring and accounting [2]
SLA Monitoring
BP Enactment
Other extensions
Gateway-based infrastructure [1]

Figure 7: Core functionality and possible extensions. The arrows indicate dependencies (A is necessary for B) and the numbers indicate prioritisation.

The basic demonstrator exercises important functionality from the following TrustCoM sub-systems:

- EN/VO infrastructure, Trust and Security Services, Policy Services – all required for the integrated infrastructure and access control
- SLA Management – lower-level aspects of SLA management to do with performance monitoring are relevant to Fundamental Monitoring and Accounting
- VO Management – required for Basic EN Membership and Service Management

Highest priority was given to **Fundamental Access Control** because:

- without a system of flexible, fine grained, context-sensitive access control that is manageable from the perspectives of individual partners and the EN and VOs the concept of agile/dynamic VO is not really viable;
- this aspect of the TrustCoM reference implementation is well-developed and embodies a number of innovations that are ripe for exploitation;
- it is potentially exploitable as a network-hosted secure resource virtualisation and exposure service even without other aspects of the VHE.

Second priority was given to **Fundamental Monitoring and Accounting** because:

- it provides the basis for a wide variety of higher-level services concerned with valuable functions such as accounting for the purposes of billing, SLA compliance monitoring, load balancing, performance monitoring, anomaly and fault detection, detection of fraud and other forms of untrustworthy behaviour;
- this aspect of the TrustCoM reference implementation is less well-developed than access control as a flexible generic capability, but has been explored in the more specific form of examples of performance monitoring in the context of SLA compliance monitoring within the testbed scenarios of Action Line 2;
- It is not reliant on Fundamental access control and is potentially exploitable independently of it. Note however that it is dependent on the same gateway-based infrastructure as Fundamental Access Control.

Third priority was given to **Basic EN Membership and Central Service Management** because:

- Some form of membership and service management at the EN level is necessary in conjunction with either or both of the higher priority functional sub-systems to provide support for an EN (i.e. a community of service providers) rather than to individual service providers. Note that central service management is distinct from management of service exposure by individual partners, which is a basic function of the infrastructure and required for all aspects of VHE functionality;
- A VO Management subsystem (aka VO Toolkit) is available as part of the reference implementation. It provides management capabilities covering the lifecycle of a 'typical' TrustCoM VO together with a graphical user interface. A lot of adaptation would be required to entirely adapt the toolkit to the requirements of the VHE and the AS scenario, which features an EN with many of the features of a 'classic' VO plus dynamically assembled and instantiated VOs. Some aspects of the toolkit provide more advanced functionality than required for entry-level EN support, while more basic EN management functionality is not supported. Furthermore management of the VO lifecycle in the AS scenario needs to be largely automated.
- It is dependent on the preceding functional sub-systems in the sense that it adds value to them by providing an extra of management functionality.

Implicitly, all three of the above are dependent on the gateway-based **EN/VO Infrastructure**, which also provides the flexibility to evolve VHE capabilities through, e.g. deployment of new and enhanced services on gateways.

3.3 Gateway architecture

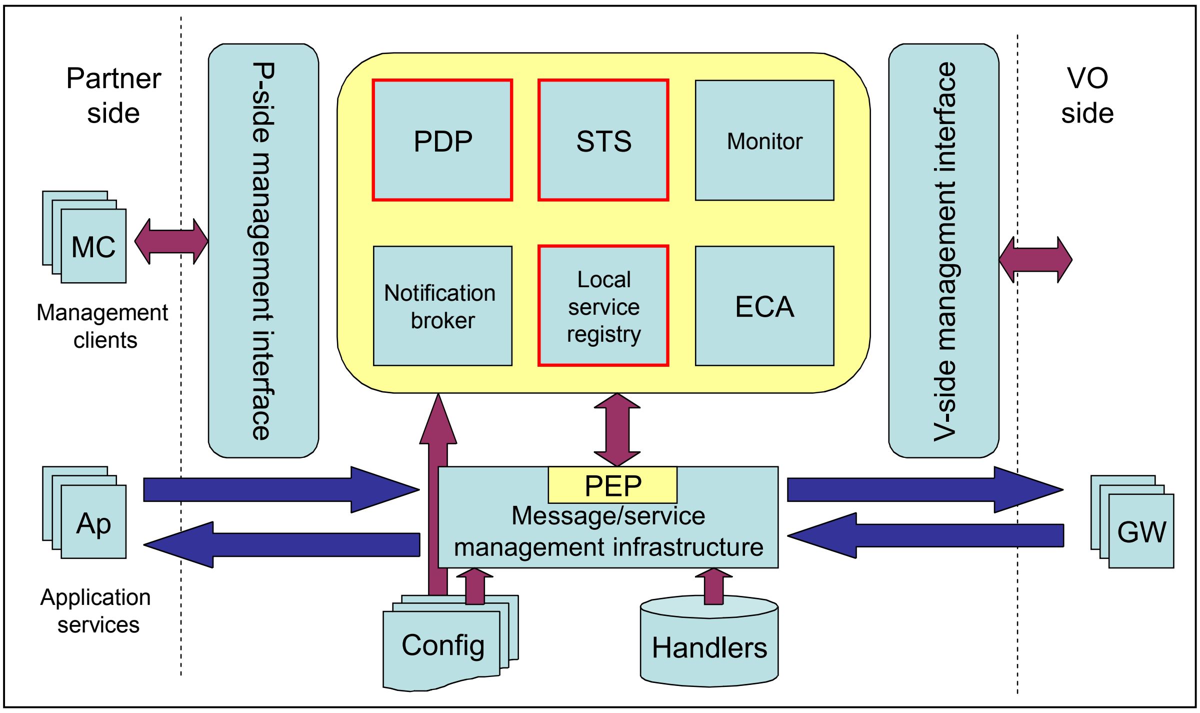

Figure 8 Gateway architecture and interfaces

The Gateway is the primary modular building block of the VHE infrastructure. It packages together a number of generic TrustCoM services on a 'per-partner' basis, as shown in Figure 8. The lower central part of the figure shows the local element of the message / service management infrastructure. Messages to and from application services of the partner are intercepted and operations are applied to them at the Policy Enforcement Points (PEP) before they are forwarded to the destination service, another gateway, or else dropped. The operations applied and their sequence depends on the message content and context as specified in configuration files. The operations are implemented mostly as independently re-usable 'handlers', thus one way of extending the functionality of gateways is by adding to its library of handlers.

In order for a partner service to send or receive messages via its gateway, it must be 'exposed' within one or more collaboration contexts, referred to as federations. This requires instantiation of a local service identity at the gateway for each federation with which the service is exposed. These service instances:

- encapsulate the application services, controlling information about them released outside the partner enterprise including real service identity and location.
- manage state for the service in the context of individual collaborations.

Messages sent to or from application services that have no corresponding exposed service instances will be blocked at the gateway. Messages between exposed service instances will only be forwarded in the context of a federation to which both services instances belong.

Information about the service instances is held in the Local Service Registry, shown within the rounded box in the upper part of Figure 8. This box contains an extensible set of modular components used within the gateway. The components shown in the figure and described briefly below provide core functionality required for Fundamental access control and Fundament monitoring and accounting capabilities. They also exemplify how the gateway could be extended to provide additional capabilities. In most cases their functionality is invoked from the message/service management infrastructure or in response to events. An event notification is a type of message that is delivered on a 'publish-and-subscribe' basis. Notification-aware components are connected by a system of event buses. A notification producer posts notifications on an event bus when e.g. a significant state change occurs in an entity it is monitoring; notification 'consumers[4]' listen on the bus for notifications of interest to them. The event notification service is a basic infrastructural element, complementing the service-service message infrastructure and similarly it is used in a variety of contexts. As well as forming part of the control mechanism of an individual gateway it can be used to coordinate between gateways and to underpin a variety of higher-level services. To preserve modularity and separate event notification traffic of different types and sensitivities, we have taken the decision to have separate event buses for intra- and inter-gateway notifications.

The highlighted elements in the rounded box in Figure 8 are directly involved in providing the fundamental access control. The Security Token Service (STS) is concerned with issuing, validating, exchanging, etc. tokens (i.e. credentials suitable for attaching to messages). The Policy Decision Point (PDP) makes access control authorisation decisions based on declarative policy rules. Both roles are well understood and accepted (at least at an abstract level) and governed (though not completely specified by) open standards (e.g. WS-Trust, SAML, XACML). The STS and PDP are invoked from handlers performing operations such as (STS) replacing 'internal' credentials recognised within a partner organisation for 'external' ones recognised within an EN, and (PDP) passing or dropping messages based on access entitlements.

The Event Condition Action (ECA) policy engine is used to implement aspects of gateway control functionality (including aspects of access control), but is also a useful generic building block with applicability to a wide range of services including notification and monitoring. As its name suggests it processes declaratives rules of the form 'If an event matching template E is observed and conditions C apply, then perform action A'.

The notification broker adds value to the basic event notification mechanism. Rather than monitoring a notification bus continuously, a notification consumer may subscribe to a topic via a broker. The broker will inform all subscribers to a topic when events concerning that topic are posted.

The element labelled Monitor is representative of a class of component concerned with gathering, correlating processing, logging and/or forwarding information from 'sensors'. Typically the sensors are either implemented as handlers (collecting information derived from messages transiting the gateway) or by instrumenting elements of the VHE or application services with extensions able to generate notifications corresponding to significant events. Examples of uses to which this class of component would be put include monitoring for the purposes of logging billable events, anomaly detection, load balancing, and compliance with SLAs. We cannot implement a comprehensive range of monitoring capabilities in the time available for the demonstrator. Instead we will implement a small number of simple representative capabilities with general

[4] This is the term that is commonly used, but it is a misnomer, as notifications are not removed from the bus when read.

applicability to the monitoring of parameters concerned with simple transactions (e.g. response time, completed / incomplete transactions, etc.).

The two remaining aspects of the gateway as depicted in Figure 8 are the Partner-side (P-side) and VO-side (V-Side) management interfaces. The P-side interface provides a means for authorised administrators within the partner organisation associated with the particular gateway to control and configure the gateway. In practice, the P-side interface is a collection of web-based administrative interfaces enabling different tasks to be performed. An example task to be performed via this interface is exposure of an additional service to the EN.

The V-side interface has two aspects:

- a gateway-gateway 'control plane' interface enabling activities such as formation of federations
- a gateway-EN/VO management interface allowing agreements, policies, decisions and events at EN and/VO level to influence the behaviour of the gateways

3.4 Central EN / VO management services

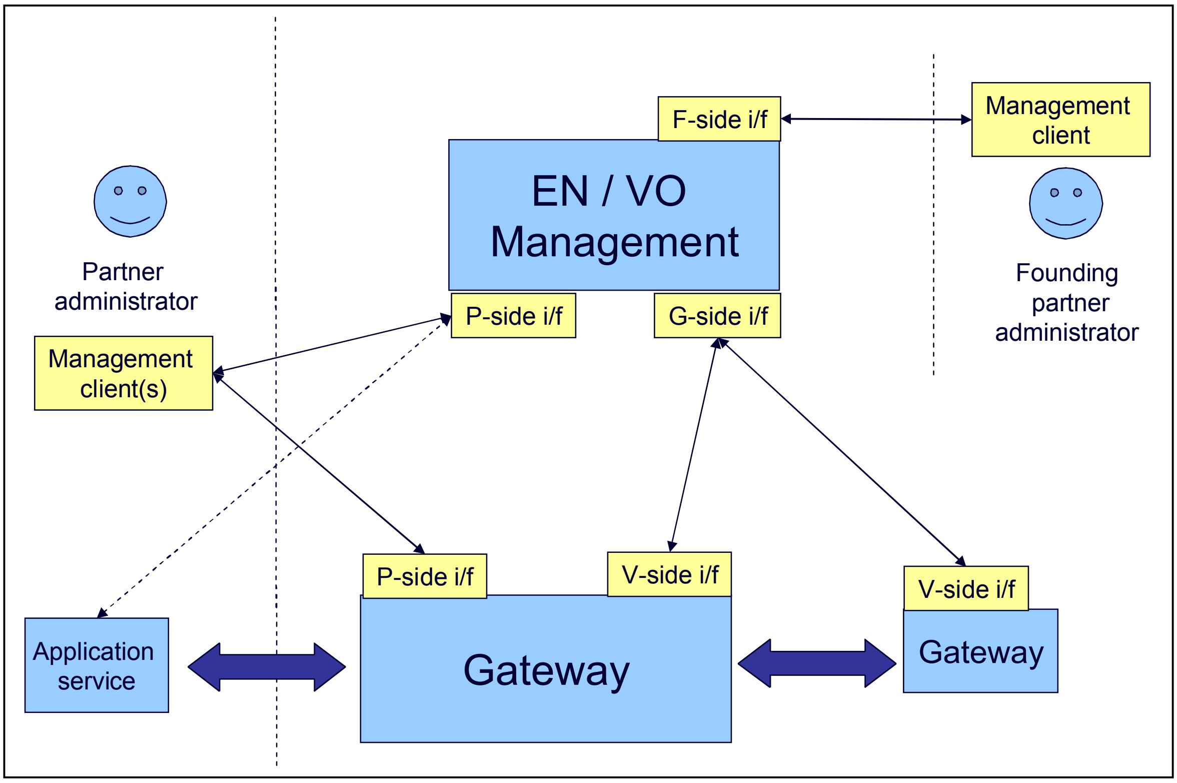

Figure 9: EN/VO management interfaces. In the interface (i/f) labels, P stands for partner, V for VO / EN, and G stands for gateway)

This element of the demonstrator packages together services implementing the central aspects of EN and VO management support provided by the VHE. Recall that the overall architecture can be represented by three stacked planes: data plane, control plane and management plane (see Figure 6). The control plane essentially consists of a federated collection of partner-specific gateways as discussed above, plus optional EN-wide services discussed below. EN/VO Management lies (primarily) in the management plane. Compared with the gateways, it deals with higher level issues concerning the EN and VOs as entities. The most fundamental of these are:

- EN Lifecycle and Membership Management: keeping track of what enterprises are members of the EN, providing means for members joining and leaving, providing means of establishing and disbanding the EN, etc.
- Central Service Management: keeping track of what services have been contributed to the EN by each partner, their characteristics, constraints on their availability, etc. The main purpose of this is to facilitate the identification of services matching desired characteristics and the formation and operation of federations and VOs.
- VO lifecycle management: Keeping track of active VOs within the EN, providing means of establishing and disbanding the EN, etc.

The above description implies the need for repositories holding information records describing: the EN, the EN member enterprises, services available with the EN, and at least the active VOs. In many cases, there are associated data structures in the control layer representing the same or related entities on a more technical level. Thus, services are represented both in the gateways and in the management plane, but the information held in the control and management plane structures is different, reflecting the different roles of the planes. Similarly, the VO (a management plane concept) is related to the control plane concept of a federation. The two are not simply representations of the same entity at different levels, however. A federation is basically an interaction context for a collection of exposed services, including policies governing permitted interactions. In the VHE, no services may interact outside the context of a federation. Consequently, a VO must be associated with at least one federation, otherwise the application services within the VO could not interact. However, in principle there may be multiple federations 'within' a VO. There may also be federations that are not associated with a VO, for example federations allowing control and management services to talk to each other. Furthermore, a federation may function as a lightweight substitute for a VO in appropriate situations.

Figure 9 shows the central EN/VO management element and its interfaces to other entities. Interaction with the:

- Founding Partner Administrator concerns EN lifecycle issues[5].
- Partner Administrator concerns a) membership issues e.g. the partner joining or leaving the EN, b) issues to do with services contributed by the partner to the EN c) issues to do with VO lifecycle management. Note that in the type of dynamic application of interest here, many interactions regarding VO lifecycle management need to be automated. In some cases it will natural for the partner side of the automated interaction to be handled by an application service. In the diagram, this is symbolised by the dashed arrow.
- Gateway primarily concerns maintaining associations and consistency between information held at the management plane and control plane levels. This includes: a) representation of services and service instances b) VOs vs. federations c) commitments to partners implied by VO membership vs. access control policies

[5] The Founding Partner is considered to be an EN member with additional duties and privileges to do with the EN as entity.

4 Demonstrator implementation

This section describes how the VHE architecture covered in the preceding section has been implemented in the demonstrator. First it outlines how the architectural elements are distributed across the partner sites, then an account is given of what has been implemented in each of the main architectural groups, including an explanation of developments relative to the Reference Implementation.

4.1 Deployment architecture

The concrete architecture for the demonstrator reflects the abstract testbed architecture shown in Figure 1 in quite a literal way. The main host locations for the demonstrator are BT at Adastral Park, Ipswich, UK, and Atos Origin, Barcelona. Servers at Atos host the scenario-specific application services, and BT hosts the VHE elements. This distribution ensures that we do have a clean separation between the application services generic platform, which is a key aspect of the VHE concept. Thus, BT represents the VHE operator's sites, and Atos represents the various EN partner sites. A gateway instance is provided at BT corresponding to each EN member in the scenario. All communication between the application services belonging to different EN members goes via the gateways of both partners.

SAP has also deployed copies of the EN/VO Central Management services at its site in Karlsruhe, Germany. As well as having some practical advantages (SAP is responsible for this aspect of the demonstrator), it allows us to demonstrate distributed deployment of the VHE. Other variations were also considered and would have been demonstrated but for shortage of time, specifically the following possibilities:

- One or more of the gateways could be hosted by Atos to represent the case where the gateway is sited on customer premises.
- One or more gateways may be technically dissimilar and hosted by Atos to represent the case where an EN member provides its own gateway.
- One or more EN member sites may be hosted at BT to represent the case where application hosting is outsourced to a hosting service provider who is also the VHE operator.

4.2 Gateway implementation

The gateway implementation for the WP38 demonstrator is based on the gateway from the reference implementation in AL2. The main alternative considered was to base the WP38 gateway on a leading Commercial Off the Shelf (COTS) web service security gateway or Enterprise Service Bus (ESB) product. The advantage in a COTS-based gateway implementation is that it would be more attractive to a commercial organisation considering becoming a VHE provider for a number of reasons including: confidence in long term support and continued development, a robust and proven technical platform and the possibility of strategic relationship over a wider product range. The main disadvantage to the COTS option is uncertainty over the flexibility and extensibility of current commercial products. After due consideration it was decided that the risks in following the COTS route outweigh the benefits. Evaluation of COTS products to examine their potential for use a basis for production implementations of the gateway is still viewed as a valuable exercise, thought due to time constraints this will take place outside the scope of TrustCoM.

The main enhancements relative to the Reference Implementation gateway and associated services are the following:

- Much greater integration of the gateway infrastructure with the primary services resident on the gateway (PDP, STS, ECA policy engine, notification infrastructure, Monitoring services, etc.) The WP38 integrated gateway is a coherent deployment unit in a modular and scalable architecture.
- Much greater automation regarding the configuration and operation of the gateway. This applies to operations initiated by the gateway administrator, by application services, and by central management services.

- Improvement of management interfaces and other enhancements aimed at creating a convincing demonstration of the VHE concept as a usable and useful network-hosted service.
- Replacement of the default PDP by one based on the Delegent product from Axiomatics AB (http://www.axiomatics.com/). Axiomatics is a commercial spin-off from SICS, the partner who developed the PDP in the reference implementation. Advantages in using Delegent include support for the new XACML 3.0 standard. It is also intended for Delegent to be a commercially supported product, which adds to its credibility as part of an eventual live deployment. Delegent was not used in the reference implementation because of timing of design decisions and potential issues with open source licensing.
- Use of notification mechanism in line with standards that have matured since commitment to certain design choices in the reference implementation.
- Event-enablement of key gateway services to allow performance monitoring and automated management using the notification mechanism and policy engine
- Use of a new version of the Imperial College ECA policy engine
- The ability for a partner administrator to replace the default STS and PDP with (compliant) ones of his chosing. In particular the STS and PDP developed by the University of Kent can be used in preference to the default ones from EMIC and SICS/Axiomatics.

4.3 EN/VO Central Management implementation

The EN/VO Central Management services in the demonstrator are adapted from the corresponding elements of the reference implementation, namely the VO Management toolkit Host, Initiator and Member editions and the GVOA manager (which conceptually can be regarded as a constituent of the Host edition). These services deal with VOs only and not ENs, but it should be noted that the standard TrustCoM notion of VO lies somewhere between the WP38 notions of EN and VO. This is because in WP38, we are dealing with simple, often short-lived, dynamically generated VOs. Consequently the management activities (such as agreement of contracts) that need a lot of time and human involvement need to be associated with the EN rather the VO. Individual VOs in the EN, specialise and implement decisions made at the EN level. Thus, in WP38, the EN can be thought of as an abstract VO without executable business processes, while the VOs proper are lightweight entities without e.g. full-blown GVOAs. Consequently, the VO Management services are candidates for aspects of both EN and VO management in the demonstrator.

The components (editions) of the reference implementation VO Management services are as follows:

- Host Edition: this provides core services including maintenance of the information repositories and databases associated with VOs in the EN. It makes use of a UDDI repository holding EN member and service information, the GVOA manager looking after contracts corresponding to VOs, a lifecycle manager responsible for maintaining information about the state of VOs, and a membership manager that provides all functions related to participant management in a VO (e.g. adding, removing or replacing members). The host edition provides a simple, browser-based graphical user interface mostly for simple monitoring purposes.
- Initiator edition: this interacts with the partner who initiates the creation of a VO and assists it in managing the lifecycle of the VO, including finding and selection of VO members/services. The initiator edition provides a comprehensive, browser-based graphical user interface tailored to the specific needs of a administrator initiating a VO.
- Member edition: this helps potential participants registering their business and services in UDDI (host edition) so they can be identified as a potential VO members by a VO initiator. It provides management functionality to enable members to confirm or reject invitations for a VO and also other performs higher-level management and configuration functions on a per-partner basis. The member edition provides a comprehensive, browser-based graphical user interface tailored to the specific needs of VO member.

Mapping these components to the VHE architecture, one instance of the Host Edition is deployed centrally. The Initiator and Member Editions act as clients to the Host Edition and have a browser-based user interface. They can be deployed at the partner sites as thick clients or centrally and accessed over the internet in thin-client fashion. Each EN member requires access to a Member and each EN member able to initiate the formation of aggregated services needs access to an Initiator edition.

While there is a good match between the requirements of the VHE Central EN/VO management function and the reference implementation VO Management, there were a number of issues requiring adaptation of the VO Management Toolkit to the requirements of the VHE. The issues included:

- Aligning the content and format of the data structures maintained by the VOM Toolkit and those of the gateways to effective coordination of operation of the VHE at the management and control plane levels
- Provision of additional APIs to support interaction between VOM Toolkit and gateways and between VOM Toolkit and application services
- The Reference Implementation VOM Toolkit incorporates support for business process management that is not part of the VHE core functionality. The toolkit was adapted to allow the required functionality to be used independently of the business process functionality.

The main enhancements relative to the Reference Implementation of the VO Toolkit and associated services address the interaction between the central EN/VO management and the gateways of the VHE as well as ease of use of the graphical user interface:

- Central Service Registry: The Host Edition was enhanced to include a central service registry database and API. Conceptually, this database enhances the more general UDDI registry already provided. Gateways and application services in the VHE use central service registry to coordinate the process of instantiation and provisioning of virtual resources.
- Collaboration Definition and Member Search: collaboration definition and member search algorithm were tailored to the specific needs of the AS scenario.
- Federation Creation: The security federations in the gateways are established according to the interaction pattern specified in the collaboration definition of the scenario.
- Policies: Policy templates are pushed to the gateways directly by using a new interface.
- Administration: Host and Member Editions GUI were revised for easier administration of UDDI entries and Business Cards.

4.4 Other EN-wide services

This section covers utility services available centrally for use in VOs formed within the EN. Typically, these will be independent of the application, but in a real VHE some services may be specialised to a particular industry sector. This class of services serves as a major means of expanding the functionality provided by a VHE when used in conjunction with the flexible, policy base facilities provided by the gateways. In the time available to TrustCoM it has only been possible to deploy a small number of basic services as examples of this class:

- Secure Audit Web Service, acting as a trusted repository to which events and documents can be sent for future reference, e.g. in case of disputes.
- A central monitoring console and ‘complex event processing’ engine that combines a number of performance monitoring, accounting, logging and analysis functions, albeit in a simple form. This is representative of a broad class of more specialised services that would be useful in a VHE.

4.5 Application services

The original application used in the WP38 demonstrator is derived from the AL2 AS testbed, which is set in an eLearning context. The main services involved are:

- an eLearning portal providing end-user access to bespoke learning services combining material from multiple content providers. There is a back-end to the portal that implements a web service interface for communication with other partner services via its VHE gateway.

- Services providing access to learning resources (i.e. course modules) owned by different providers. These deliver course content to be integrated by the portal.

The second application used, is based on the Virtual Music Store scenario introduced above. The main services are:

- Front end services (music store clients) customised to particular virtual music stores allowing end users to select content to down load

- Content-providing services representing (e.g. genre-based) catalogues of music copyright owners or their representatives

These services are deployed over Apache MUSE [16], an implementation of the Web Services Resource Framework (WSRF) standard. Otherwise they are unconstrained in implementation other than that they are expected to comply with a collection of widely accepted web services standards, mostly in the WS-* family.

Use of MUSE was found to improve interoperability among the different web services and also to enhance their implementation. The applications required binary data to be transferred within SOAP messages encoded as text. This was a challenging test for interoperability as it involves some of the newer aspects of the standards used. One major benefit is that it allows the whole message (including the binary payload) to be signed and/or encrypted. The practical expertise gained in this exercise has been documented and would be a significant help to any enterprise developing services in the TrustCoM way.

5 Demonstrator functionality

This section describes how the demonstrator implements the VHE core functional services: Fundamental Access Control, Fundamental Monitoring and Accounting, and Basic EN Membership and Central Service Management. The functionality is explained by means of a set of use cases spanning the lifecycle of a VHE-based enterprise network. The use case descriptions actually cover more of the lifecycle than is supported by the demonstrator in order to provide a context for the implemented functionality and also point the way towards extensions that could usefully be implemented after the end of TrustCoM to bring the demonstrator closer to a fully-functional prototype platform. The accounts of the use cases indicate where steps are not supported in the demonstrator. Before covering the use cases, we revisit the application scenarios in a little more detail.

5.1 Application scenarios

5.1.1 Generic scenario

A typical SME-based scenario would be the following: An operator (say BT) offers a ('virtual') hosting service to SME enterprise networks, i.e. it is the VHE operator and hosts kernel services to which the participating SMEs federate their resources (see Figure 2). A customer organisation approaches BT with a view of establishing an SME EN in a particular business domain (say). The EN is 'instantiated' with an initial core membership (maybe just the founder) and opened for business. SMEs approach the EN management to join the EN. The joining process involves various things like federating the new SME's platforms with the EN core platform, agreeing to contracts, registering services, and so on. A customer comes along with a particular requirement, and a set of partners and services are selected from within the EN community, etc. The VO is formed and interacts with the customer to agree and deliver services.

5.1.2 E-learning scenario

We originally selected eLearning as an example business domain for the enterprise network. The main categories of partner in the eLearning EN are:

• **Training Consultants** – TCs interact with end-users to understand their training requirements and formulate them in a way that may be used to construct a personalised training programme.

• **Training/eLearning providers** – these act as integrators, building bespoke training packages and co-ordinating their delivery to the end-user

• **Content providers (aka Learning Resource providers)** – these modular resources that may be used with training packages

• **The eLearning EN operator** – provides additional services such as a specialised portal, payment services via banks, etc. as well as generic services supporting trust, security and contract management in the operation of the VO.

All of these are members of the EN The eLearning EN operator has the special status of EN founder. The people requiring training are the external consumers.

A typical use case might go as follows:

The end user accesses the portal and selects a training consultant. This might be one the user has an existing relationship with, or might be one specialised in a particular topic or in a particular category of user. The training consultant interacts with the user to obtain training requirements, then issues an invitation to selected training providers to offer bespoke course that meet the requirements. The training providers respond with programmes constructed from modules offered by various content providers. Advised by the training consultant, the user selects an offer from one of the training providers. The selected training provider coordinates the delivery of the training as required by the user. The training consultant may maintain an involvement in this phase to monitor the user's progress and advise on changes to the programme. At the end of the course payments are distributed to the various service providers subject to user satisfaction and fulfilment of obligations.

5.1.3 Virtual Music Store scenario

In the Virtual Music Store (VMS) example, the main categories of partner are:

- The music EN operator
- Music content providers, e.g. record labels or other copyright owners
- Virtual on-line music store operators

A virtual on-line music store is seen as serving a specialist market (e.g. jazz enthusiasts). It is likely to provide additional content (such as blogs, reviews and recommendations) aimed at its target market. The music store operator reaches agreement with appropriate content providers to re-sell selections from their overall catalogues. The virtual music store is a VO consisting of the music store operator and content providers with which it reaches re-sale agreements.

In this case the VO is specified by the VMS operator using a graphical user interface (GUI), rather than being constructed on the fly by an application. The provisioning of the VO is done automatically.

5.2 Key Stakeholders and actors

Table 1 below enumerates the main stakeholders in an EN hosted by a VHE and the advantages for them of the VHE concept in comparison to a bespoke EN infrastructure. A less formal account was given earlier in Section 2. Table 2 is a similar listing of actors that appear in the use cases. Typically the actors in Table 2 are active representatives of the stakeholders in Table 1 in the context of the use case.

Stakeholder	Business Function	Description
VHE operator	the entity hosting the enterprise network and providing re-usable infrastructure and core services	This is a new business role, though related to application and datacentre hosting. For a NGN operator such as BT, it is an opportunity to leverage existing network infrastructure to provide added value, higher margin services. There is a high degree of synergy between the VHE concept and the move to a service oriented approach to exposing generic network 'capabilities' taken by some network operators.
EN prime / founding partner	the entity running the EN. It may also be a service provider/consumer within the EN. It is similar to the VO initiator in other TrustCoM scenarios.	This is assumed to be an entrepreneurial entity or one motivated by a wish to serve or promote the formation of a co-operating community. It is likely that such an entity would have problem domain knowledge or business expertise as its core skill. Existence of a VHE would enable it to launch and operate an EN with little capital investment. It would also require relatively little technical expertise (though this depends to some extent on the facilities provided by the VHE operator).
EN member	a service provider and/or consumer within the EN community	The VHE concept allows the EN members to outsource the support for co-operation. The outsourcing occurs at several levels. The most basic is that of the infrastructure for secure and trusted interoperation of services with other VO members. In addition, integration of the VHE platform with an

		underlying 'capability-oriented' NGN infrastructure would allow member services to draw upon a wide range of network-based services (e.g. billing platforms). In the eLearning scenario Training Consultants, Training/eLearning providers and Content providers (aka Learning Resource providers) are examples of EN member. The eLearning EN operator is both a prime partner and EN member. In the VMS scenario, the VMS operators and music providers are EN members. One of the VMS operators may act as the EN prime partner, or else this role is played by a distinct entity.
External consumer	an entity external to the EN benefiting from services provided by VOs formed within the EN. This could be an employee within an EN member company or an external customer.	Benefits from increased confidence in the services provided by the EN. Not only is there more confidence in the technology, but trust is to some extent inherited from the VHE operator's brand. Integration of the VHE with the underlying network platform is likely to give a more seamless experience – consuming services provide by the EN is little different for consuming other services from or hosted by the network operator. In the eLearning application scenario, the person with need for training is the external customer. In the VMS scenario, the purchasers of the music tracks are the external customers.

Table 1 – Key stakeholder listing form

Actor	Role	<u>Description</u>
VHEoperator	Administrator of the VHE	This actor works for the VHE provider. The role amalgamates various aspects of administration of the VHE facility and individual VHE instances, and interacting with representatives the VHE provider's customer (the EN prime partner).
ENFoundingPartner	Administrator for overall the Enterprise Network	This actor works for the EN prime partner. The role amalgamates various aspects of administration of the Enterprise Network as a consortium, configuring the VHE to serve the EN, and interacting with representatives of its service provider, the EN provider,
ENPartnerAdminstrator (for partner X)	Administrator acting on behalf of an EN member.	This actor works for a service-providing EN member. The role amalgamates aspects of administration of how services belonging to this partner are exposed within the EN, management and monitoring of other EN/VHE controls available to the partner, and interaction with the EN management function.
ApplicationService/User	PartnerEN member-owned service interacting with others	This actor is a software service owned by an EN member and made available for interaction with services offered by other partners on a client and/or provider basis. The service may be acting on behalf of a user (external consumer), including the case where the service is a relatively thin user agent or portal allowing a user to

	within the EN	interact as a service with other services. Note that the other listed actors are specialisations of ApplicationService as they two are services interacting via the same mechanisms
AggregatingService	Application service able to select partners to form an aggregated service	This actor is an application service that is able to form an aggregated service by recruiting appropriate supplier services form within the EN. In the VMS scenario there is no AggregatingService as such. The role is played by the administrator of a VMS using a VO management user interface.

Table 2 Listing of key actors in the use cases

5.3 Lifecycle stages

The main lifecycle use cases are as follows:

- EN is founded / disbands
- Partner joins/leaves EN
- Partner exposes/replaces/withdraws service and configures associated policies
- VO / composite service created or disbanded
- Composite service executes, policies are applied and monitoring and accounting actions performed
- Monitoring and accounting cases operating over longer timescale

These will now be considered in more detail. A standard presentation format is used for each use case. First the actors appearing in the use case are listed, an informal account of the main steps in the use case is then given, finally comes a more detailed account of events in the demonstrator within the use case.

5.4 Found/disband Enterprise Network

5.4.1 Actors

- ENFoundingPartner
- VHEoperator

5.4.2 Use Cases:

This covers set up and dissolution of the EN. Founding the EN involves the founding partner deciding to form an EN, reaching agreement with the operator to provide a VHE. The operator then instantiates an ‘empty’ VHE which is configured by the founder. Disbanding the EN covers the reverse process at the end of the EN’s life. The main constituent sub-use cases are described below in terms of typical narrative flow.

AgreeToHostEN: The ENFoundingPartner has decided he/she wants to establish an EN in a particular domain (e.g. eLearning), but does not want to operate and maintain the technical platform for it and so approaches the VHEoperator. The two parties agree on contractual terms and other information affecting what facilities the VHE needs to provide.

InstantiateVHE: The VHEoperator instantiates the VHE and allocates resources required for its operation. This could involve creating a new VHE instance on dedicated servers in a server farm, on one or more virtual machines, or allocating resources on a VHE platform capable of hosting multiple ENs. The ENFoundingPartner is provided with access to the VHE to allow him/her to configure it to support the specific EN.

InitialiseEN: The ENFoundingPartner configures the VHE to support the specific EN. Tasks in principle include defining:

- a framework contract / agreement that EN members must sign up to,
- domain-specific service and notification topic ontologies/taxonomies
- a set of VO templates appropriate to this EN,
- configuring central registries that will hold records on EN membership, available services, VOs, etc.

At the end of the life of the EN a complementary set of activities take place to disband the EN and release to VHE resources. Note that it may be necessary for some vestigial records and services may need to be retained beyond the formal end of the EN.

5.4.3 Support in demonstrator

AgreeToHostEN: This is assumed to take place via a separate channel. No support is provided.

InstantiateVHE: In a real service, the ability to instantiate a VHE to serve a new EN easily and quickly is important. No direct support for automated instantiation is provided in the demonstrator, however.

InitialiseEN: In the context of the demonstrator, this primarily involves customising, configuring and/or initialising a number of registries or repositories associated with the VOM toolkit to the application context of the EN. The main registry functions associated with the VOM Toolkit are:

- Business (UDDI) registry: this holds business information on the EN members and the services they offer within the EN. The service descriptions here are used for e.g. discovery and selection of members and services for inclusion within a VO.
- VO registry: this holds descriptions and necessary technical parameters of the VOs existing within the EN (e.g. name/objective of a VO, roles and associated partners etc.)
- Central service registry. This contains technical information on services required e.g. for service execution)

Specifically, the following must be done:

- Define a framework GVOA applicable to all VOs within the EN and upload this to the toolkit
- Define service/role and VO type taxonomies to be used as a basis for classification, description and naming of services, roles, choreographies and templates within the EN.
- Define and upload choreography descriptions. In the VOM Toolkit, choreographies specified in CDL are used to describe VO structures in terms of roles and relationships. The roles in the choreographies are later matched to service providers able to fulfil these roles as part of the VO formation process.
- Decide the information allowed / required to be entered for the partners and service descriptions in the business registry and configure the registry/information entry console appropriately.
- Decide any non-standard fields required/allowed in the VO and central service registries, and configure the registries appropriately.

Active support for these activities in the demonstrator is limited. For WP38, the management of business registry entries was enhanced as it is considered crucial for demonstration audiences to appreciate the standards-based UDDI approach. Moreover, the roles a member is willing to play in an EN can also be defined over the GUI. Otherwise the activities are part of the 'back story' of the demonstration scenario, i.e. they are not demonstrated explicitly, but are understood to have been performed before-hand.

5.5 Partner joins/leaves EN

5.5.1 Actors

ENFoundingPartner

ENPartnerAdministrator

5.5.2 Use Case:

This use case occurs each time a partner joins or leaves the EN. It is quite likely that blocks of partners will join soon after the EN is established and leave as part of the disbandment process, however, partners may also join or leave the EN at any time during its active life. The use case is essentially the same in each case.

The partner reaches agreement with the founder to join the EN. This will involve agreeing to the terms of the EN framework contract/agreement amongst other things. Information about the partner is captured and stored in the EN membership registry. Various provisioning activities are then performed to give the new partner access to the VHE. These include instantiating a gateway for the new partner, providing the ENPartnerAdministrator with administrative access to it, and recording its details in the appropriate central registry. The ENPartnerAdministrator then configures the new gateway. The gateway's local identity management service also needs to be added to the federation linking the all the partner gateways in a trust network.

Complementary tasks are performed when a member leaves the EN.

5.5.3 Support in demonstrator

It is useful to distinguish between EN management-level activities performed by the ENPartnerAdministrator primarily via the VOM Toolkit and technical-level (infrastructure) activities performed by the ENPartnerAdministrator primarily via the management console of the gateway serving this partner. The ENFoundingPartner does not have an active role in the demonstrator, though in a real scenario he/she would need at the very least to accept or reject a request from the partner wishing to join the VO.

Management level:

The ENFoundingPartner provides the new partner's ENPartnerAdministrator with access to VO toolkit Member and (optionally) Initiator editions. These are clients communicating with VOM Toolkit host edition. These can either be instances dedicated for use by that partner and installed on a partner-owned machine, or else be central resources accessed over the internet via a browser. The partner is identified by a name and a unique business key.

The new partner's ENPartnerAdministrator:

- reviews the Framework GVOA and must confirm agreement or else cancel the registration. The simplified scenario supported by the demonstration does not allow negotiation of partner-specific contractual terms.

- enters the description of his/her employer in the business registry. This includes providing 'business card' that is used later in establishing trust relationships with other partners in a VO.

Technical level:

A new gateway, with characteristics and configuration defined by a gateway profile, is instantiated for the new partner. Information to allow custom configuration is gathered from the ENPartnerAdministrator. The gateway profile references a default infrastructure profile type defined by the new partner at this time e.g. specifying a choice of PEP.

The gateway PDP is deployed with a trusted root policy which says that a (later issued) policy is valid if the issuer of the policy is the ENPartnerAdministrator of the service owner. Root policies are not changed during operations in the EN.

Management-technical interface:

A reference to the Federation Manager (part of the gateway) is added to the VOM Toolkit membership configuration

5.6 Partner exposes/replaces/withdraws service and configures associated policies

5.6.1 Actors

ENPartnerAdminstrator

5.6.2 Use Case:

A partner nominates an existing service to be exposed within the EN. Information on the service is captured and local and central service registries are updated. There is a complementary activity enabling a service to be withdrawn.

Following declaration of a service, there is a collection of activities to do with setting policies and configuring the mechanism controlling how a service is exposed. It must be performed at least once before a newly-added service can be used, but may also be performed at other times during the life of the service, e.g., when policies change. Sub-activities include:

- Setting/updating access control and other policies concerning this service that are not VO-specific.
- Updating various local information repositories used by services such as the STS and PDP.

5.6.3 Support in demonstrator

Again, we distinguish between EN management-level activities and technical-level (infrastructure) activities.

Management level:

The ENPartnerAdministrator enters descriptive information about the service in the business (UDDI) registry. Note that the name used to describe the service type must adhere to the conventions established in service taxonomy to ensure consistency with the roles used in the choreography descriptions and to enable the business registry entries to be matched up with the corresponding services exposed via the gateway. The relationship between them is as follows. The service name/description entered in the business registry denotes the ability to play a role as defined in the choreography description. The service capability exposed at the gateway actually provides the capability to play the role. For reasons of expediency, the simplistic convention adopted in the demonstrator is that the same name must be used in all three places. The way this works is described below when discussing VO formation.

Technical level:

To expose service via the gateway, the partner administrator defines the new service capability, giving it a name in line with the conventions established earlier. He/she defines a connector that links exposed capability with 'real' services. For generality, a tag is attached indicating whether to push information to the VOM toolkit.

Setting policies and configuring the mechanism controlling how a service is exposed is a technical level activity performed via the gateway. Tasks include:

- Setting service access control policy templates and claims. When a partner exposes a service, the service is associated with the owner and the correct ENPartnerAdministrator. This means that the PDP root policy will accept service access control policies set by the ENPartnerAdministrator. Note that actual access control policies apply to a service instance within a VO/federation. The templates and claims provided here and others that are tied to roles in a VO are used later to create the policies.

- Setting service-specific monitoring policies: These are essentially event-correlation rules used to used e.g. to generate accounting events or performance metrics.

Management-technical interface:

The Gateway registers the service with the central service registry. The record includes a role name identifying the capability as the one to use when the partner plays the named role in a VO, and a reference that can be used later to create an instance of the capability ('instantiator EPR').

5.7 Create/disband VO / aggregated service

5.7.1 Actors

AggregatingService or initiating ENPartnerAdministrator

5.7.2 Use Case:

This is the use case for forming a light weight / ad hoc VO (aggregated service) within the EN. It involves selecting and agreeing appropriate partners / services and performing other activities required before enactment of the corresponding business process. Note that even a simple consumer-provider pair could be considered as an elementary aggregated service. In terms of the TrustCoM VO lifecycle it combines the VO identification and VO formation stages. A complementary activity takes place at the end of the aggregated service's life.

In the VMS scenario, the VMS operator wishes to create, e.g. a specialist on-line jazz store so searches for providers of jazz catalogues within the enterprise network. The jazz VMS operator and the selected catalogue providers become the VO. The VMS operator's portal/client plus the catalogue services that provide access to the music tracks become the aggregated service.

In the eLearning scenario, an end user interacts with a learning portal which translates the user's learning requirements into a course plan or 'learning path'. The portal then selects providers of services corresponding to requirements for modules identified in the learning path. The portal operator plus the selected module providers become VO. The portal plus the services providing the modules become the aggregated service.

5.7.3 Support in demonstrator

Again we distinguish between EN management-level activities and technical-level (infrastructure) activities.

Management level:

The AggregatingService (e.g. eLearning portal) or initiating ENPartnerAdministrator (e.g. acting for a VMS operator) interacts with the VOM Toolkit to create a record describing the new VO / aggregated service. This happens as follows in the case where the process is driven by the initiating ENPartnerAdministrator using the VOM Toolkit Initiator Edition:

1. The administrator is prompted for a name and objective for the VO and other 'header' information about the particular VO (VO ID, start date, etc.). Note that the VO description being generated must later be correlated with the corresponding technical-level template. In the current demonstrator this is done by matching the text in the VO objective.

2. the administrator specifies the structure of the VO by selecting a choreography description from a list presented by the VOM Toolkit.

3. the administrator selects files containing access control policy templates (i.e. incomplete XACML policies) and claims and monitoring and accounting policies to be applied within the VO. In the current demonstrator, these monitoring and accounting policies apply only at VO level, being enacted by a central policy engine.

4. the VOM toolkit searches the Business Registry to identify EN members offering services that match the roles in the choreography. This is done based on the service name/description entered in the Business Registry by the ENPartnerAdministrator during the 'partner exposes service' step. In the current demonstrator, the partners offering services with names matching the role names in the choreography description are selected as candidates to play the roles in the VO.

5. For each role in the choreography, the administrator is presented with a list of EN members able to play the role, from which he/she selects at least one.

6. the VOM toolkit Host Edition sends invitations to the administrators of the potential partners via their instances of the VOM Toolkit Member Edition

7. the administrators of the potential partners accept or decline the invitations as appropriate.

8. the VOM toolkit Host Edition stores a record representing the VO and containing the above information in an internal database (the VO registry).

9. the VOM toolkit Host Edition passes information to the appropriate gateways to allow them to provision the VO. This is described below.

An analogous process is followed in the case of a VO created by an aggregating application service, except that interactions are between application services and VOM Toolkit editions via a web service or other API. Logic must be encoded within the application services to enable them to specify appropriate choreography descriptions, policy files, etc., to select partners from candidates able to play given roles, and to respond appropriately to invitations.

Management-technical interface:

No communication is required during VO identification. During VO formation one or more calls are made by VOM toolkit to each participating gateway passing the following information:

- Access control policies and claims. Role-specific access control policy templates (i.e. incomplete XACML policies) and claims are pushed to the appropriate gateways.

- The potential exists to pass role-specific monitoring policies, but this is not currently done.

- Business cards of partners with whom trust relationships are needed. The choreography description used during VO formation defines bindings between roles, indicating e.g. client-provider relationships. The business cards are exchanged in order to establish the trust for mutual recognition as e.g. authorities signing security assertions.

- VO ID,

- VO type: this is a string obtained from the "objective" field as entered by the EN partner administrator. It is used by the gateway to identify the appropriate infrastructure profile to use.).

Any ECA policy specified (actual policy not template) is pushed to an EN level policy engine.

The VOM Toolkit host edition provides a view of a VO generated from information spread across its various registries including the following information:

- VO ID (this acts as a key)

- associations between roles and instantiator EPRs

- associations between instances and instance EPRs. The instance EPRs are initially null/empty. The information is pushed by the gateway when a new instance is created.

Technical level:

Triggered by the call from the VOM toolkit, a federation view is created and an infrastructure profile for the VO is configured based on a template file. The template to use is identified by the VO type information passed in the call. This happens at each gateway corresponding to a partner in the VO. The term 'federation view' refers to the local record held in a gateway describing a federation. The local views are linked by a federation id derived from the corresponding VO ID. Each gateway adds its parent business entity and other VO members to its federation view.

5.8 Execute composite service / operate VO

5.8.1 Actors

ApplicationService/User

AggregatingService

5.8.2 Use Case:

At the end of the end of the previous step, a federation corresponding to the VO had been created, but no services had been instantiated within it. Service instantiation can take any time after creation of the federation and before the service instance in question is used. Services do not necessarily have to instantiated at the same time as each other, and instances can also be deleted / replaced at any time during the life of the federation.

In the demonstration scenarios considered here, a VO consists of an initiating partner providing an aggregating service, and other partners acting as sub-contractors to it and providing constituent services. The aggregating service provides the composite service to an end-user, and interacts both with the end user (via a graphical user interface) and with the constituent services.

In the VMS demonstration application a distinct instance of the Music Store Client service is created for each user of a VMS. This acts as a personal aggregating service for that user. This allows the TrustCoM mechanisms to be used to provide access control on a per-user basis – e.g. ensuring that a user can access only a particular set of tracks. The constituent services (corresponding to e.g. genre-based catalogues) are instantiated when the VMS is used by its first customer. Subsequently, a new user-specific client each time a new user uses the VMS's services.

5.8.3 Support in demonstrator

The following is step-by-step account of events in the context of the VMS scenario:

1. Before opening for business, the VMS operator uses application software to send a service instantiation request for each of the catalogue services in the VO to the relevant gateways. These specify the service type/role, the VO ID, the URI for the new instance. The EPRs to send the requests to for each role can be obtained by querying the (central) VOM toolkit host edition using the VO ID allocated to the VMS during VO identification as a key. Note that it is possible to tell if a service has already been instantiated within a VO by querying the (central) VOM toolkit host edition. If the service has not been instantiated, the EPR for the given service instance will be null.
2. In response to these requests the virtualization service of the relevant gateways create service instances that are specific to this VO (i.e. this VMS).In the course of creating a new service instance, the virtualization service of the relevant gateway calls the corresponding STS, PEP Management Service, PDP, and the ECA Adaptation Engine. Each of these services is then individually configured. In addition,

the virtualization service publishes information on the new service instance to the VOM Toolkit Host Edition.

3. The VMS operator publishes a web page for a portal service by means of which customers can access the VMS, and the store is now open for business.
4. Some time later, a jazz enthusiast visits the web page for the first time. He goes through a registration process, and the portal service requests a client instance specific to this user. It does this be sending a request to the VMS operator's gateway specifying the service type/role, the VO ID, the URI for the new instance and information identifying the user. This is analogous to instantiation of the catalogue services, but note that a client is instantiated within the federation for each individual user.
5. The user is presented with a selection of tracks to download by the personal VMS client. He selects one or more is these and in response, the download service is invoked.

More details of what happens during service download are given below. First we elaborate on the generation of access control policies. The access control policies that apply to a service within the context of a VO are a result of combining default policies (applying to all services protected by a given PDP), service specific policies applicable to all uses of this service, and policies applicable to the service when playing a particular role in a particular VO. The last two categories are derived from the policy templates and claims respectively specified when the service is exposed, and pushed to the gateway by the VOM toolkit when the VO is formed. These are instantiated and completed at the gateway with the authority of the ENPartnerAdministrator during service instantiation and installed in the PDP with the appropriate precedence relative to other policies. In the absence of VO-specific templates pushed by the, the ENPartnerAdministrator issues an access control policy which grants the use.

The following is describes security-related events that happen in the course of a typical request made by a client application service to a provider application service. The main internal actors involved in the story are:

- The client and provider application services
- Intra-enterprise credential authorities in the two application service providers. These issues X.509 certificates identifying the application services within the corresponding enterprises.
- The two gateways corresponding to the two application service providers
- Two Security Token Services (STSs) installed at the gateways. These issue and check (SAML) security tokens used in inter-enterprise communications within a VO. Due to the prior exchange of business card, the two services accept the assertions in each other's tokens.
- Two Policy Decision Points (PDPs) installed at the gateways. These hold the XACML access control policies derived from the service- and VO-specific templates and claims mentioned above. They interpret these policies to answer queries as to whether a message should be allowed to proceed.

The major steps in the chain of events are as follows:

1. **Send message**: The client application service sends a request in the form of a SOAP message with a header conforming to the WS-Addressing standard. An X.509 certificate identifying the sending service within its parent enterprise is attached to the message. The message is routed initially to the gateway corresponding to the parent enterprise of the sending service.
2. **Access control on outgoing message**: The gateway receives and parses the message. It sends an XACML request to its local PDP asking whether the sender identified by the X./509 certificate is permitted to request the action requested in the context of the given VO. The PDP applies the policy rules and replies that request is allowed.
3. **Credential exchange (outbound):** The gateway the sends a request using the WS-Trust protocol to the local STS to exchange the intra-enterprise X.509 certificate for a SAML assertion recognised within the relevant VO.
4. **Gateway-gateway message forwarding:** The message with VO-specific tokens signed by the STS attached is then forwarded to the gateway corresponding to the service provider. The message integrity (and optionally confidentiality) is protected using the WS-Security standard.

5. **Token validation:** The receiving gateway asks its local STS to validate the token attached to the message. This is successful because of the trust relationship established previously between the STSs. The response includes validated claims and the intra-enterprise X.509 certificate of the target application service.
6. **Access control on incoming message**: The gateway then sends an XACML query to its local PDP asking whether the requested action is allowed in this VO context. The request includes the X.509 certificate of the destination service and the validated claims of the requester.
7. The validated message is then delivered to the target service.

A similar sequence of events also occurs during delivery of the response message.

5.9 Monitoring and accounting functionality

5.9.1 Actors

The following actors are stakeholders with an interest in setting monitoring policy or receiving the results of its application:

- VHE operator (monitoring the effective functioning of the platform)
- ENFoundingPartner (concerned with functions relating to operation of the EN as a whole (including audit logging and accounting for the purposes of revenue sharing and billing)
- Initiating partner of a VO (concerned with effective operation of a VO, e.g. detecting exceptions indicative of problems, SLA violations, etc.)
- ENPartnerAdministrators as customers of other services (e.g. wanting information on trustworthiness of partners, reliability and performance of services, SLA compliance, etc.)
- ENPartnerAdministrators as providers of services (e.g. wanting to monitor the effective performance of their own services and gateway, usage by others, etc.)

5.9.2 Use Case:

We are concerned here with the management of Monitoring and Accounting functionality used to provide information for the following basic purposes:

- assessing reliability / trustworthiness of EN members as service providers
- measuring and monitoring (performance) parameters of the sort that could appear in SLAs
- information to be used as input to billing and payment clearing services
- management of the gateways and overall VHE platform

It is useful to distinguish a number of levels of granularity, including:

- service invocations (records of successful and unsuccessful invocations, including response time and reference to transaction context) and lifecycle events (e.g. deployments and instatiations, creation of VOs / federations, etc.)
- transactions (involving a provider supplying a service to a 'consumer') - records of completed, unsuccessful (i.e. transaction lifecycle was initiated but transaction did not actually take place for various 'normal' reasons), and failed transactions (including reference to VO context)

- VO / end-end lifecycle of a composite service involving multiple members - similar issues to transaction, but more complex.

Generally speaking, the monitoring and accounting sub-system consists of the following types of element:

- ‘sensors’ detecting basic events
- Information processors that correlate and add value to more basic event information and generate events of higher granularity and information content
- Information sinks, typically information stores, analysis functionality and/or consoles
- Notification infrastructure that takes events produced by sensors and information processors and delivers them to the subscribing information processors and sinks. In the VHE architecture, we have two distinct levels of notification infrastructure: the lower one deals with notifications within the sphere of influence of an individual EN member, the upper one send notifications between partners at an EN or VO level. Certain information processors are able to bridge between the two levels.

The principle management use cases for the monitoring and accounting sub-system are:

- Design-time configuration of components to give them the potential to act as producers / consumers of specific types of event
- Operational configuration of components to determine their actual behaviour as consumers and producers. This may be context dependent, e.g. apply within a given VO or VO type.
- Configuration of the notification infrastructure in terms of a network of publishers and subscribers
- Setting of policies governing the behaviour of the information processing elements.

5.9.3 Support in demonstrator

Demonstrator components are instrumented to allow them to generate the following types of basic event:

- Successful delivery of a service. This could be an application or infrastructure service. The event structures differ for the two types.
- Failed delivery of a service
- Violation of QoS guarantees / SLA / other constraint in the course of delivering a service
- Component overload (
- High volume of service request denials
- High volume of service access attempts
- Mismatch between claimed and measured response times
- Lifecycle events

The following information sinks are implemented:

- Monitoring console
- Diagnostic log

- Audit log
- A repository of billable events notionally accessible to a billing service

The following intermediate correlation and information processing is performed:

- Generation of aggregated billing records from more primitive events. This includes combining service delivery events with SLA violation events that may result in penalties.
- Generation of alarms conditions from more primitive events, and in some cases trigger an adaptive response action. Examples of alarms include: overloaded components, suspected denial of service attacks, anomalous behaviour such as excessive access denial events, persistent SLA violations
- Comparison of self-reported performance figures (e.g. response times) with those measured at gateways e.g. in order to keep partners honest.

6 Subjective evaluation

6.1 Overall assessment

Delays in other TrustCoM work packages resulted in a highly compressed timescale for WP38. While we have successfully implemented the planned core functionality, at the time of writing, it has not been possible to give any demonstrations and hence to receive feedback from interested parties outside the project (or indeed outside the WP38 team). Consequently, what is given here is a subjective evaluation of the demonstration by the team involved in producing it.

The main objectives of the demonstrator were to:

1. Make progress towards commercial exploitation of TrustCoM framework
2. To create a prototype VHE, i.e. a platform based on the TrustCoM framework that a provider (e.g. BT) can use to offer services to VOs.
3. Evaluate the potential for integration with service oriented 'capabilities' exposed by IP-based Next Generation Networks (NGNs), e.g. BT's 21st Century Network (21CN)
4. Show commercial potential via a convincing example application.

Certainly, the first two have been achieved to a meaningful level. Experience during development of the demonstrator has reinforced our view that the VHE concept and business model are very relevant to emerging business requirements and well-aligned with the business strategies of TrustCoM partners. Furthermore, availability of production implementations of the VHE will open up opportunities for innovative small businesses to experiment with new business models, products and services, which is a key objective of the FP6 programme. See Section 2 above for more explanation of these points.

The demonstrator definitely qualifies as a useful early prototype for a VHE service platform. It implements a set of functions that are core to the role of the VHE, and the modular architecture allows for easy extension of functionality through addition and upgrade of services. Furthermore, the core functions represent a significant advance over the state of the in secure exposure and rapid integration of software services in an inter-enterprise environment. While much work would still be required to turn the demonstrator into a production service platform, the key features are in place and have been shown to function as advertised.

The third point has not really been addressed through practical experimentation due to shortage of time. However, the demonstrator provides a suitable vehicle for proceeding with this in the future. The availability of BT's Web21C SDK will facilitate integration of access to NGN services as additional central services within the VHE architecture. It is in the TrustCoM partners' own business interests to continue this investigation beyond the end of TrustCoM and the BT Web21C team is keen to co-operate.

The fourth objective has been achieved to a degree. The two example applications studied are suitable to help explain the business concept and technical requirements and for providing a easily appreciated context for demonstrating the VHE prototype. They are not in themselves 'killer applications' for the VHE platform. However have essential characteristics that will facilitate engagement with key stakeholders in order to 'brainstorm' killer application ideas with them.

6.2 Choice of core functionality

It was always realised that a fully-featured prototype VHE platform was not achievable within the scope of TrustCoM. Indeed, such a platform would need functionality beyond the TrustCoM scope of trust, security and contract management in VOs. Consequently, the plan and architecture described in [15] provide for expansion of capability around a modular and infrastructure framework and core functionality. The core functionality was identified as: Fundamental Access Control, Fundamental Monitoring and Accounting, and Basic EN Membership and Service Management. This functionality has all been implemented and experience in developing the demonstrator has confirmed this choice of core functionality to have been a good one. All three areas are distributed sub-systems involving central services at EN and/or VO level and 'per-partner' services integrated with the gateways. They integrate well with each other via the distributed infrastructure, and are mutually supportive. The overall package does seem to be a good, minimal mix of capabilities for a VHE. Additional functionality could be provided as add-on modules or integrated with the application services, but loosing any one of the three core functions would impact the utility of a VHE significantly.

Some aspects of the TrustCoM Framework we deliberately omitted from the VHE core functionality or are included in a more fundamental form. These decisions seem to have been justified by experience so far. Full SLA Management and also Reputation were replaced by the Monitoring and Accounting sub-system. This has enabled us to provide a much more general and flexible capability upon which are range of useful functions (including SLA Management) can be built. It is based on event reporting, notification, event correlation, and event-condition-action processing engines. Since it is programmable via policy rules it can easily be adapted to moe specific requirements within an EN. It has been used to demonstrate some simple but effective functionality in the areas of application performance monitoring, VHE management and performance monitoring, accounting (for the purposes of billing and revenue sharing), anomaly detection, and audit logging. Much of this is relevant to SLA Management and supplier selection based on track record in any case. The higher level functionality lost relative to the Reference Implementation / testbeds demonstrated previously, is in any case problematic for practical business application in the near term: Electronic representations of SLAs and associated penalties or other sanctions are not yet sufficiently mature or standardised to allow automated derivation of monitorable performance constraints or actions applying the sanction from realistic SLAs. Automated application of sanctions on SLA violation can also be problematic on legal grounds.

Support for Business Process Enactment was also omitted. While some form of coordination of service execution is typically required within an aggregated service, whether the co-ordination is better performed by the application services themselves, or by application neutral middleware services depends on the nature of the application and design philosophy adopted for the application services. Experience with the application scenarios studied confirms that the coordination logic is often quite closely tied in with the application logic. In the particular type of VO structure adopted it was most natural to incorporate the co-ordination logic in the 'aggregating' application service. This acts as a 'prime contractor' that combines input from a number of subsidiary services to produce a composite output. Note that WS-CDL is still used as a means of describing the VO structure, a purpose to which it seems well suited.

6.3 Holistic requirements

Qualitative requirements established for a VHE include:

- Ease of management of the EN and its infrastructure by an EN operator that does not have technical expertise
- It must be quick and easy to set up new VOs
- There must be a balance of control between enforcement/detection of non-fulfilment of obligations and autonomy of the partners
- The VHE must support multiple simultaneous VO with overlapping membership, which means access control and other mechanisms must be flexible and context dependent

Substantial effort was put into the demonstrator to enhance manageability and integrated operation of the architectural components. While there is always room for improvement, significant progress has been made to this end. The basic mechanisms are in place and have been shown to work. In particular, the complex operations concerning management of the integrated gateways, virtualised services and federations are now highly automated. Coupling between this 'control plane' level of operation and the 'management plane' embodied in the VO Management toolkit has also been improved greatly. Areas where further improvements can be made in terms of ease of management include: integrated management consoles for the various administrative roles, and tools for specifying and editing policies, templates, etc.

Balance of control is largely achieved by separation of responsibilities between the gateways and the central management functions (VO Management Toolkit). Gateway policies and behaviours are specified by the administrators working for the corresponding partners, while the VO Management Toolkit deals with issues across the EN or the VOs within it. VO-level access control policies are divided according to role within the VO and distributed to the gateways for enforcement. Since the gateways are under the control of the partner administrators, then the final say on access control lies with the EN partner whose service is affected. We feel that this is how things should be. Even if a partner has entered into an agreement to allow access to a service, it should not be forced to do so by the technology. Instead, apparent violations of agreements should be detected and raised for considered by the EN management. Having said this, conflicts of VO and partner policies and the mechanisms for resolving them have not really been explored within the demonstrator and application scenarios.

The final point, multiple simultaneous VO with overlapping membership, is a major strength of the demonstrator and a key distinguishing feature and selling point for the demonstrator as a VHE prototype. The application services implemented allow this capability to be demonstrated, but do not really allow its full potential to be explored.

6.4 Extensions beyond the core functionality

The functionality that has been implemented in the demonstrator is core functionality identified in [15]. As explained earlier, the architecture allows for this to be extended incrementally. This sub-section documents some suggestions for useful developments to the demonstrator beyond TrustCoM, either to make progress towards an initial production VHE, or to extend the functionality. Note that the issues documented here are not (on the whole) research topics, but relate to bringing the demonstrator closer to being a ‘first of a kind’ service platform prototype.

Migration of the infrastructure to a COTS appliance- or software-based platform: Offering a VHE as a hosted service requires a highly-reliable, commercially supported technology base able to support a high volume of transactions and multiple independent ENs. At the moment, the demonstrator’s technology base is still open source and/or experimental software. As two complementary ways forward in this regard, BT is in the early stages of experimenting with migration of the gateway infrastructure to a commercial web services gateway appliance, and integration of the gateway with an industry-strength COTS enterprise service bus product. Related to this is the need to extend

Support for VHE provisioning: This is related to the previous point. It concerns the ability for a VHE operator to create rapidly a VHE instance as a platform for a new EN (see InstatiateVHE sub-use case above in 5.4**)**. This platform could be ‘real’, i.e. consist of software and/or hardware dedicated to the new EN, or else ‘virtual’, i.e. appear to be dedicated to the new EN but in fact share resources with other ENs. The same issues apply to provisioning of gateways to new EN partners. Related to both of these is the question of whether an enterprise that participates in more than one EN can use the same gateway for all the ENs.

Customisation to an application domain: This concerns support provided to the founding partner of an EN in adapting the generic VHE to the application domain focus of the EN (see InitialiseEN sub-use case above in 5.4). This includes defining domain-specific service and notification topic ontologies/taxonomies that establish vocabularies for describing service capabilities and for access control and monitoring and accounting policies. Outside TrustCoM, the semantic web community and various industry bodies are active in defining generic formal frameworks and standardising vocabularies in partcular industry sectors, respectively. It is hoped that such work could provide the basis for a toolset for the VHE platform. Another example of domain customisation is the definition of VO templates (i.e. a generalisation of the choreography description files in the VO Management Toolkit). One approach is to assemble a library of generic templates as part of the VHE. The EN founder could then specialise these templates to create VO structures and choreographies that were characteristic of VOs is the EN’s domain. These could then be selected and adapted during VO/aggregated service creation as in 5.7 above.

Management of contracts: The current demonstrator allows an EN contract to be stored and presented to new EN members for agreement. Tools could be provided to help an EN founder to craft an appropriate contract by selecting and adapting optional clauses e.g using a configurator-style tool or an expert system embodying the knowledge of legal experts. Going beyond this in the direction of automated enforcement or assurance of contractual terms is likely to be difficult without considerable progress in the formal representation of legal contracts. It should be possible, however, to provide tools to guide a human manager in deriving enforceable or observable conditions from contracts.

Assistance in writing policies: Three sorts of policy languages are used in the demonstrator: XACML for access control policies; PonderTalk, the language of the new Event-Condition-Action policy engine from Imperial College for adaptation, and an Even-Condition-Event policy language for monitoring and event correlation. All require a combination of domain knowledge and technical expertise in order to write policies. In an EN, such policies would need to be written by managers / administrators in the partner companies who would not have the specialist technical expertise. Tools are therefore required to assist them in writing the policies, analysing them for consistency, etc.

Improved and integration of management consoles: Manageability of the platform from the point of view of view of managers and administrators in the EN partner companies will be a key factor in a successful VHE platform. Substantial progress has been made in this direction. The gateways, the VOM Toolkit and certain other components (e.g. the monitoring sub-system) provide management interfaces for human administrators. However, these need to be developed further and also integrated so that an administrator in a given role is not aware of using different tools for different tasks.

Improved ease of integration with application services: This mainly involves provision of a richer set of APIs by means of which application services can interact with gateway and central management functionality.

Monitoring and accounting-based functionality (including SLA Management): The capability of the current demonstrator in this area is basic, but useful. It provides an understandable example of how the mechanisms implemented in the demonstrator can be used to create a variety of useful higher-level functions related to performance management, anomally detection, billing and revenue sharing, detection of SLA violation, track record (for partner qualification) etc. To exploit this infrastructure to its full there need to be means of a) configuring it easily to requirements of an EN, and b) allowing the partner and EN administrators to modify its behaviour during the operational phase (e.g. asking to be notified when certain constraints are violated. In addition, more built in support for specialist would also be useful. This would facilitated by Providing a library of standard handlers for use in the gateways, especially handlers that are easily configurable to detect and report message-related events on a selective basis.

Business Process Enactment: As remarked above, this was present in the Reference Implementation of the VOM Toolkit, but deliberately omitted from the core functionality of the demonstrator. The VOM Toolkit derives BPEL descriptions of parts of the overall VO business process from the choreography description and sends these to the appropriate partner for enactment. In the VHE-based scenarios studied here, there are three main destinations for the BPEL descriptions. Firstly, The BP engine could reside on the partner company sites, and logically be part of the application services. In this case, the VHE does not enact the business processes, but may need to communicate about choreography so that the application services understand the constraints they must comply with for the VO's business process to function smoothly. In the second case, a BP engine could be integrated with the gateways to coordinate the application service execution via the virtualised services. The third case involves a central BPEL engine that either choreographs the overall VO business process (e.g. acting on behalf of the VO initiator) or subsidiary processes primarily involving central services.

Extended central EN and VO management functionality. In addition to specific topics mentioned above (e.g. contract management) most extensions to VHE functionality will require commensurate extensions at the central management level.

Additional central services. One of the main mechanisms for extending the functionality of the demonstrator is by adding central services. Examples that could be added post TrustCoM include:

- Suitably wrapped billing and disbursement engines
- Services recording historic performance of EN members and making performance-based measures available e.g. in support of partner selection
- A recommender service allowing VO participants to submit favourable or unfavourable reports on experiences of doing business with other EN members and to obtain reputation measures based consolidated reports
- Electronic notary service
- Flexible and programmable / configurable interface onto the EN-wide distributed monitoring functionality
- Services offering monitoring and management of performance and configuration of the VHE infrastructure

- Services encapsulating capabilities offered by the underlying network. See the following section for the examples currently available via the BT Web21C Software Development Kit.
- Services offering access to wider, federated identity infrastructure.

NGN integration The VHE is envisaged as a facility that is integrated with a next-generation, converged, multi-service network (NGN) such as BT's 21st Century Network (21CN). The VHE provides a way for the NGN operator to offer higher level, added-value services to its business customers. Conversely, the VHE can make use of the NGN service-oriented capabilities (e.g. authentication, billing, etc.) as well as the ability to integrate communication services with application services, and use of the network to send web service messages. The ability to demonstrate the potential for integration and later to trial integrated operation would enhance considerably the benefit of the demonstrator to BT. It had been intended to perform experiments to confirm the feasibility of this and investigate the best strategies for integration to allow work to continue smoothly after the end of the project and to support the credibility of the exploitation route. Regrettably, time did not permit this, but the desirability of it still remains.of performing trial integration/interoperation with one or more BT 21CN 'capabilities' (essentially service-oriented interfaces to network services). The Web 21C SDK mentioned above and its developments should facilitate this.

Genuine business pilot application: The two applications studied in WP 38 serve their purpose in providing examples that are easy to understand and that can be used to explain the main ideas behind the VHE and their implementation in the demonstrator. They do not, however, fully exercise the more sophisticated capabilities of the VHE. A good follow-up to TrustCoM would be to develop the current demonstrator further as the basis of a larger-scale trial in conjunction with an end-user organisation with a significant unmet need in the EN founder role.

7 Conclusions

Despite delays in other TrustCoM work packages that resulted in a highly compressed timescale, WP38 has achieved the majority of its objectives. We have produced a prototype VHE embodying the core functionality identified in [15] and that implements the TrustCoM Framework in a form suitable for operation as a hosted service. The prototype is well suited to demonstration of both the business concept and the technical features that derive business advantage from technical results produced within TrustCoM. Two example applications have been implemented, one set in the eLearning domain, and the other concerning virtual music stores. They are suitable to help explain the business concept and technical requirements and for providing an easily appreciated context for demonstrating the VHE prototype. They are not in themselves ‘killer applications’ for the VHE platform. However, they have the essential characteristics that will facilitate engagement with key stakeholders in order to ‘brainstorm’ killer application ideas with them.

The demonstrator is a genuine implementation of the TrustCoM framework, employing all its functional sub-systems with the exception of Business Process Enactment. This is not part the core functionality for a VHE, but is seen as an optional feature that may be used in some contexts, but not others. The main reason for this decision is that in many cases, business logic is interwoven with application logic and is best integrated with application services rather than via a central facility. Choreography descriptions are still used as an important part of the VO management process, however. Full SLA Management is replaced by an Monitoring and Accounting sub-system. This has enabled us to provide a much more general and flexible capability upon which are range of useful functions (including SLA Management) can be built.

The vast majority of services featuring in the TrustCoM Reference Implementation have been further developed and used in the demonstrator, specifically: the VO Management Toolkit, Gateway message processing and policy enforcement Infrastructure, XACML PDP (from SICS/Axiomatics), STS (from EMIC), an alternative PDP and STS from Kent, ECA policy engine (substantially re-written), Secure Audit Web Service, etc. The Monitoring and Accounting sub-system is largely new, though drawing on ideas and experience from the Reference Implementation. Relative to the Reference Implementation, the TrustCoM services are much more closely integrated within an overall architecture. This applies especially to the Integrated Gateway, which combines gateway infrastructure with services provided on a per-partner basis (PDP, STS, ECA, etc.). Substantial effort was put into the demonstrator to enhance manageability and integrated operation of the architectural components. In particular, the complex operations concerning management of the integrated gateways, virtualised services and federations are now highly automated. Coupling between this ‘control plane’ level of operation and the ‘management plane’ embodied in the VO Management toolkit has also been improved greatly.

Deployment and integration of the demonstrator was challenging due to the use in combination of many emerging open standards whose implementations are not yet mature and prone to interoperability problems. On the whole, diligence paid off and we have eventually been able to make the technology work as we wanted it to. As a particular example, use of MUSE (an open source implementation of the WSRF specification) was found to improve interoperability among the different web services and also to enhance their implementation. The applications required binary data to be transferred within SOAP messages encoded as text. This was a challenging test for interoperability as it involves some of the newer aspects of the standards used. One major benefit is that it allows the whole message (including the binary payload) to be signed and/or encrypted. The practical expertise gained in this exercise has been documented and would be a significant help to any enterprise developing services in the TrustCoM way.

Compared to other service aggregation platforms, the TrustCoM VHE’s main added value lies in its ability to deal with virtual organisations as overlapping business collaborations between autonomous enterprises rather than just providing facilities for integrating loosely-coupled services. The ability to support multiple simultaneous VO with overlapping membership and to provide flexible and context dependent access control and other mechanisms is a major strength of the demonstrator and a key distinguishing feature and selling point. The VHE provides a balance of control between enforcement/detection of non-fulfilment of obligations and autonomy of the partners. This is largely achieved by separation of responsibilities between the gateways and the central management functions (VO Management Toolkit). Since PEPs are located at the gateways, which are under the control of the partner administrators, then the final say on access control lies with the EN partner whose service is affected. We feel that this is how things should be. Even if a partner has entered into an agreement to allow access to a service, it should not be forced to do so by the technology. Instead, apparent violations of agreements should be detected and raised for considered by the EN management.

At the time of writing, it has not been possible to give any demonstrations and hence to receive feedback from interested parties outside the project, so the assessment of the demonstrator has been purely subjective and from an insider’s point of view. Experience during development of the demonstrator has reinforced our view that the VHE concept and business model are very relevant to emerging business requirements and well-aligned with the business strategies of TrustCoM partners. Furthermore, availability of production implementations of the VHE will open up opportunities for innovative small businesses to experiment with new business models, products and services, which is a key objective of the FP6 programme.

The demonstrator is a valuable vehicle for ‘industrial’ partners in TrustCoM to gain commercial benefit from the project’s results. It will be used widely to stimulate a dialogue with business representatives over novel product and service propositions. There is also always a demand for demonstration of advanced technology to the customer community to provide thought leadership, and also to elicit feedback that will help define product roadmaps. Activities such as this are a necessary pre-cursor to commitment of the investment required for development of a prototype production platform. Even if such investment does not materialise the insight gained is highly valuable if it helps to shape strategic direction and product and service roadmaps. It is difficult to quantify the benefit of research projects that do not lead directly to products or services, but plant ideas in peoples’ minds that influence a range of new projects. Nevertheless, the value is real.

A next logical step along the road to full commercial implementation of the concept is to conduct a genuine business pilot in conjunction with an end-user organisation acting in the EN Founder role, i.e. one seeking to establish and run a forum acting as a marketplace for software services. One can envisage a variety of motivations for this, including: profit from fees on usage charges, stimulation of innovation, efficiency of operation within a complex enterprise, and so on. Identification of a partner for such a business pilot is a key objective of the demonstration phase mentioned previously.

EU and related collaborative initiatives are possible vehicles for such a business pilot. Indeed TrustCoM partners are active in a number of FP6 projects and FP7 proposals that apply technology related to the WP38 demonstrator. In particular, the FP6 project BEinGRID incorporates a large number of business pilots including BE09, which is set in the on-line gaming domain. This has been proceeding in parallel with WP38 and has evolved a related VHE concept using a similar integrated gateway as the main architectural building block.

References

1. Two Scenarios for 21st Century Organizations: Shifting Networks of Small Firms or All-Encompassing "Virtual Countries"?', Laubacher et al, http://ccs.mit.edu/21c/21CWP001.html
2. 'Flexible Work Arrangements and 21st Century Worker's Guilds', Laubacher and Malone, http://ccs.mit.edu/21C/21CWP004.html
3. 'Electronically-enabled free lancing', *Harvard Business Review*, Sept-Oct 1998
4. Elance web site: http://www.elance.com/
5. guru.com web site: http://www.guru.com/)
6. hotdispatch.com web site: http://www.hotdispatch.com/.
7. http://code.google.com/: 'Google Code' web page that encourages development of applications and web pages using Google services by making available API's and development toolkits.
8. http://open.bbc.co.uk/labs/: BBC Innovation Labs web page. The BBC Innovation Labs is a pilot project to develop innovative new products and services with independent New Media companies from across the UK.
9. http://www.amazon.com/AWS-home-page-Money/b/ref=gw_br_websvcs/002-2500083-6650457?%5Fencoding=UTF8&node=3435361&pf_rd_m=ATVPDKIKX0DER&pf_rd_s=left-nav-2&pf_rd_r=0Y2AJNKG02RY9FBHBMHP&pf_rd_t=101&pf_rd_p=285790601&pf_rd_i=507846 Amazon Web Services page.
10. Web21C SDK Developers' Centre: http://sdk.bt.com/
11. 'Microsoft and BT Launch Connected Services Sandbox Competitions' http://www.microsoft.com/presspass/press/2007/feb07/02-12SandboxCompetitionPR.mspx
12. Connected Services Sandbox: http://www.networkmashups.com/Default.aspx
13. TrustCoM deliverable D62 TrustCoM Framework V3
14. TrustCoM deliverable D64: Final TrustCoM Reference implementation and associated tools and user manual
15. TrustCoM deliverable D38: Migration and Demonstration Plan for Ad Hoc Aggregated Services Demonstrator, Issue 2
16. http://ws.apache.org/muse/, Home page of Apache MUSE, part of the open source Apache Web Services project