

AlbatrOSS

IST-2001-34780



Architecture for Location Based Applications of Third generation Operation Support Systems

Document Identifier:	IST-2001-34780/LU/WP5/D11
Title of Deliverable:	D11 - Evaluation of Trial System Version 2
Deliverable Type: (P/R/L/I)*	P
Nature of the Deliverable: (P/R/S/T/O)**	R
Contractual Date of Delivery to the CEC:	31-12-2003
Actual Date of Delivery to the CEC:	8-1-2004

Workpackage responsible for the Deliverable:	WP5
Editor:	Ko Lagerberg (LU)
Contributor(s):	AlbatrOSS Consortium
Reviewer(s):	Søren Vejgaard-Nielsen (UHC), Mario Lopez Gallego (TID)

ABSTRACT

This report contains the evaluation results of the second version of the AlbatrOSS trial system. The trial system consists of three trial sub-systems, which are evaluated separately. The document describes the relevance and applicability of the requirements and business processes that are input to the sub-systems, and evaluates the scenarios, standards, technologies and tools used.

KEYWORDS

Evaluation, sub-trial, standards, technologies, 3G, OSS

© 2002-2004 by the ALBATROSS Consortium.

See <http://www.ist-albatross.org/> for further details

* Type: P:Public, R-Restricted, L-Limited, I-Internal

** Nature: P-Prototype, R-Report, S-Specification, T-Tool, O-Other

IST-2001-34780

AlbatrOSS

Deliverable D11

Evaluation of Trial System Version 2

Editor : Ko Lagerberg
Status – Version : Final
Date : December 2003
Distribution : Project
Code : IST-2001-34780/LU/WP5/D11

© Copyright by the AlbatrOSS Consortium.

The AlbatrOSS Consortium consists of:

Waterford Institute of Technology (WIT – Ireland) – *Project Coordinator*

Fraunhofer Institute for Open Communication Systems
(FOKUS - Germany)

T-Systems Nova GmbH Berkom (TSN - Germany)

Atos Origin Integration (ATOS - France)

UH Communications A/S (UHC - Denmark)

Lake Communications (LAKE - Ireland)

Lucent Technologies (LU - Netherlands)

Telefonica Investigacion y Desarrollo (TID - Spain)

Table of Contents

1	EXECUTIVE SUMMARY	6
2	INTRODUCTION	7
2.1	Structure of this report.....	7
3	EVALUATION RESULTS: 3G SERVICE CHARGING AND QOS MONITORING	8
3.1	Business process support.....	8
3.2	Input requirements.....	9
3.3	Functional Architecture.....	12
3.3.1	Customer	12
3.3.2	Mobile Portal Service Provider.....	12
3.3.3	Third Party Service Provider.....	14
3.4	SID Model.....	15
3.5	OSS Components	16
3.5.1	Portal Access Manager.....	16
3.5.2	CSC Manager.....	16
3.5.3	Billing Engine	17
3.5.4	AA Manager.....	17
3.5.5	Accounting Repository	17
3.5.6	RBS	18
3.5.7	FMA	18
3.5.8	Q3ADE IPDR Store	18
3.5.9	SLA Manager.....	19
3.5.10	Report Log	19
3.5.11	MAC.....	19
3.5.12	Metering Component	20
3.5.13	NPE.....	20
3.5.14	Restaurant Service.....	20
3.6	Implementation Technologies and Standards.....	21
3.6.1	Technologies	21
3.6.2	Standards and Recommendations	23
3.7	Deviations	26
3.8	Conclusions.....	26
4	EVALUATION RESULTS: INTER-TECHNOLOGY AND INTER-DOMAIN ROAMING.....	28
4.1	Business processes support	28
4.2	Input requirements.....	29
4.3	Functional architecture.....	31
4.3.1	Customer	32
4.3.2	Mobile Network Operator.....	33
4.3.3	WLAN Network Operator	34

4.3.4	Third Party Service Provider.....	34
4.4	SID model	35
4.5	OSS Components	36
4.5.1	Restaurant Service.....	36
4.5.2	Q3ADE IPDR Store	37
4.5.3	Rating Bureau Service	37
4.5.4	Accounting Repository	37
4.5.5	Billing Engine	37
4.5.6	AAA Server.....	37
4.5.7	AAA agent	37
4.5.8	AA Client	38
4.5.9	Interface Registry	38
4.5.10	Domain Manager.....	38
4.5.11	UL agent.....	39
4.5.12	OSA/Parlay Gateway	39
4.5.13	Network Roaming Manager.....	40
4.5.14	Roaming Contract Manager	40
4.5.15	Mobility Client.....	40
4.5.16	Network Monitoring Tool.....	40
4.6	Implementation Technologies and Standards.....	40
4.6.1	Technologies	40
4.6.2	Standards and Recommendations	41
4.7	Deviations	42
4.8	Conclusions	42
5	EVALUATION RESULTS: DELIVERY OF INTEGRATED SERVICES	44
5.1	Business processes support	44
5.2	Input requirements.....	46
5.3	Subtrial architecture	52
5.3.1	End-user	52
5.3.2	Integrated Service provider	53
5.3.3	Third Party Service Provider.....	55
5.4	SID model	55
5.5	OSS Components	56
5.5.1	Component User Interface Manager (UIM).....	56
5.5.2	User Security and Privacy Manager (USPM)	56
5.5.3	Session Manager (SM).....	57
5.5.4	User Profile Manager (UPM).....	57
5.5.5	Data Collection Analysis and Control Manager (DCACM)	57
5.5.6	Customer Reporting Manager (CRM)	57
5.5.7	Service Quality Manager (SQM):	57
5.5.8	User Subscription Manager (USM)	58
5.5.9	Process Manager (PM).....	58
5.5.10	Billing Engine	58
5.5.11	Q3ADE IPDR Store	58
5.5.12	OSA/Parlay X	58
5.5.13	Third-party services (non OSS components)	58

5.6	Technologies and standards	59
5.6.1	Technologies	59
5.6.2	Standards and recommendations:.....	63
5.7	Deviations	66
5.8	Conclusions	66
6	TRIAL 2 SUMMARY AND CONCLUSIONS	68
7	REFERENCES	69
ANNEX 1	ACRONYMS.....	71

1 Executive Summary

This document represents a milestone and signals the end of the execution phase for AlbatrOSS Trial System version 2, and reports the evaluation results of the trial. It is part of a collection of deliverables that describe the second version of the trial system:

- [D7] Requirements, business models, and scenarios
- [D8] Architecture and specification of the trial sub-systems
- [D9] Trial implementation and integration details
- [D11] Trial evaluation

The second version of the AlbatrOSS trial system is a logical continuation of the first trial. A number of concepts, scenarios, and components from the first trial have been reused and/or enhanced. A number of new components have also been developed to validate updated or new requirements (from [D7]).

From its initiation the project has been focused on the emerging business models in the evolving next generation of mobile communications environment. The final results of the activities of Workpackage Two (Market Analysis & Business Modelling), captured in Deliverable 7 [D7] clearly show that the key aspect of all the emerging models is the need for increased levels of integration and cooperation between all stakeholders. This integration is required to deliver innovative service packages, seamless network roaming and enhanced convenience to ever more sophisticated customers. Each of the subtrials evaluated in this document address this issue at some level, which validates their adherence to the business models identified.

Three sub-systems have been defined in this trial, in contrast to the four sub-systems of the first trial. The OSS components that are provided by the project partners are applied in different collections compared to the first trial, to validate and improve the interoperability of the components, and to verify the applicability of the implemented concepts for different scenarios. These sub-trials are focused on the following 3G OSS topics:

- 3G Services aggregation and dynamic management
- Service product provisioning through a portal
- QoS management
- Usage-based charging and convergent online billing
- Flexible and federated mediation of network and service usage
- User location support in heterogeneous networks
- Inter-technology and inter-domain roaming

The trial integration and execution has been performed without physically exchanging the partners' OSS components. Instead, the Internet was used to interconnect the partners' test labs. The distributed nature of this trial (test labs were at 8 locations in 6 different countries) has therefore put an emphasis on interconnectivity and interoperability. Web technologies including servlets and SOAP/WSDL were used extensively, and excellent results were achieved. The SOAP specifications and tools are still immature, but nevertheless it is a favourable interconnection technology for exchanging OSS related data.

The evaluation results captured in this document will be used as inputs to the specification of the final AlbatrOSS architecture.

2 Introduction

In this document the following sub-trial systems are evaluated:

- 3G Service Charging and QoS Monitoring
- Inter-Technology and Inter-Domain Roaming
- Delivery of Integrated Services

The evaluation assesses how suitable the OSS components are for the management of 3G and WLAN services. The trials are used to validate the requirements, business models and scenarios from [D7], and the functional architecture as described in [D8].

2.1 Structure of this report

The evaluation results are reported in a separate chapter per sub-trial. Each of the chapters 3, 4 and 5 follow the same basic structure:

1. **Business process support**, an assessment of the business processes that form the basis for the trial system. These processes are defined in [D7].
2. **Input requirements**, describing the input requirements for the sub-trial. The requirement numbers are used as a reference in a requirement-mapping table, which maps the requirements to the OSS components that address it.
3. **Architecture evaluation**, evaluating the architecture and scenarios from the perspective of the different stakeholders that are relevant to the sub-trial. The architecture evaluation has a qualitative nature.
4. **SID model**, evaluating the Shared Information and Data model that is part of the AlbatrOSS intermediate architecture [D8]. This section describes the applicability of the model for the sub-trial, what parts of the model are used or not used, and what is potentially missing from the model.
5. **OSS components**, assessing the value of each component of the overall architecture, the encountered problems during the trial, and the advantages/disadvantages related to using the component.
6. **Implementation technologies and standards**, evaluating the technologies and standards that were used to implement and integrate the sub-trial OSS components. Only the most relevant and/or less common technologies are included.
7. **Deviations**, describing whether or not the sub-trial implementation is different from the intermediate architecture as described in [D8], and the reasons for deviation.
8. **Conclusions**, giving an overall summary of the sub-trial assessment.

Chapter 6 concludes the evaluation by summarising the results of the complete trial.

Chapter 7 contains references.

Annex 1 lists the abbreviations used.

3 Evaluation results: 3G Service Charging and QoS Monitoring

The main objective of the *3G Service Charging and QoS Monitoring* sub-trial was to demonstrate the charging and billing of innovative 3G services, and measuring service quality to monitor customer SLA violations. This objective was achieved through the following steps:

- Provision of the service product via web portal.
- Generation, aggregation and correlation of service usage records,
- Rating the usage data based on predefined charging schemes,
- Generation of the service charges incurred,
- Measurement of network and service performance to ascertain end-to-end service quality,
- Using service quality information to monitor SLA violations.

The *Mobile Portal Service Provider* integrates a location-based Restaurant Service and a Smart Home Environment service, both of which are provided by *Third Party Service Providers* (3PSP). The portal itself is the single point of access, authentication and authorisation for the *Customer* (i.e., end-users Jack and Jill in the *AlbatrOSS Storyboard* [D7]). There are two concurrent business processes, namely Assurance and Billing, which are involved in this sub-trial and subject to evaluation.

To meet the demands of Assurance business process, performance data from each of the 3PSPs is required to ascertain the overall service quality for the services that the Customer uses. The performance data is sent to Mobile Portal Service Provider, where it is analysed and service quality is derived.

In order to bill the Customer, OSS components need to collect usage data from Restaurant Services and a Smart Home Environment service and data needs to be processed for the generation of charges.

These two concurrent business processes meet when the terms and conditions of customer SLA are not met, e.g., as a result of deterioration in service quality. If this happens, customer SLA violations can be reported to the Billing process, which in turn would take further actions.

Detailed explanations on the business processes involved in this sub-trial are given in section 4.2 of [D7]. The OSS functions involved are described in section 4.3.1 of [D8].

3.1 Business process support

This sub-trial supports a set of key aspects of Assurance and Billing business processes. Aspects are *performance and QoS* in the Assurance vertical of eTOM and *mediation, charging and billing* in the Billing vertical.

The sub-trial has undertaken two sub business process flows: *3rd Party Service Provider QoS and Billing* and *Mobile Portal Service Provider QoS and Billing* described in section 4.2.3 of [D7]. The main stakeholders involved in the processes are *3PSP, Mobile Portal Service Provider, Mobile Network Operator* and the *Customer*. Key requirements of these

stakeholders on OSS have been formulated from the business processes and incorporated in an overall OSS functions view in section 4.3.1 of [D8].

The process flows and the OSS functions are used as a blueprint to assemble a set of OSS components in an integrated management subsystem (or sub-trial system). The OSS components are interconnected by means of well-defined interfaces (Please see the component diagram in section 5.1.2 of [D9]). The MAC, NPE and Report Log components support performance and QoS aspects. The mediation, charging and billing aspects are supported by FMA, Q3ADE IPDR Store, Accounting Repository, RBS and Billing Engine components.

The sub-trial has successfully demonstrated that the two process flows take place in the sub-trial system. However, it should be noted that *Service Quality Management*, *Customer QoS/SLA Management*, *Service & Specific Instance Rating*, and *Billing & Collection Management* take place only in *Mobile Portal Service Provider QoS and Billing* sub process flow.

The way sub business process flows are supported by OSS components is briefly summarised here.

The 3PSPs collect service usage data and send it to one of the two available federated mediation facilities (i.e., FMA and Q3ADE IPDR Store), which, in turn, transform the usage data into the standardized IPDR format and forward these as IPDR documents to the RBS. The RBS extracts the usage information, uses a scheme to apply tariffs, generates charges and stores the charges in the Accounting Repository. The Billing Engine retrieves the rated usage records from Accounting Repository and includes them in the customer bill.

As to performance and QoS, the NPE monitors network resources operating in the Mobile Network Provider and 3rd Party Service Provider domains. It measures the performance using standardised metrics and transfers the data to Mobile Portal Service Provider. The MAC collects performance data, applies an algorithm to measure service quality (e.g., availability as a percentage value), compare the service quality value against a service quality objective. Service quality reports are generated and stored in the Report Log.

In summary, it can be concluded that the OSS components developed and integrated for this sub-trial support Assurance and Billing business process identified in the phase two of AlbatrOSS project.

3.2 Input requirements

The table below lists the requirements that have been used as a basis for the implementation of the sub-trial components. The requirements follow from [D7, ch.5] and [D8].

Requirement ID	Brief Description
SS1-1.0	<i>Convenient services access for the users.</i> Users must be able to access services they have subscribed to in a user-friendly manner, e.g. through a mobile portal which allows users to view and use services from a central point.
SS1-2.0	<i>Convenient service subscription for the users.</i> Users must be able to use a single point of access to subscribe to new services (including obtaining information about service costs).

SS1-3.0	<i>Secure login</i> A secure login mechanism must be offered to users.
SS1-4.0	<i>Measurement of service quality</i> A service administrator or an OSS component must be able to obtain service quality reports indicating service KQI (Key Quality Indicator) values.
SS1-5.0	<i>Comprehensive presentation of performance measurement results</i> A system administrator or an OSS component must be able to make use of KPI (Key Performance Indicator) values in order to generate service quality reports.
SS1-6.0	<i>Service quality report logging</i> A system administrator must be able to store service quality reports and warnings in a log and browse the log using a simple and convenient facility.
SS1-7.0	<i>Federated usage mediation mechanism</i> A 3GOSS must offer a federated usage mediation function to collect and aggregate usage data produced by services and network resources and to transfer the data to a rating mechanism.
SS1-8.0	<i>Collection of proprietary usage records</i> The federated usage mediation function must support collection of usage records in proprietary format offered by services and network resources.
SS1-9.0	<i>Adoption of a standard to structure usage data</i> Federated usage mediation mechanism must support an open, flexible record format (e.g. IPDR structure) for exchanging usage information.
SS1-10.0	<i>Usage measurement</i> The mediation component must be able to reconcile differences in the ways service usage is measured and recorded (e.g. time-based, volume-based, event-based).
SS1-11.0	<i>Secure authentication of network user</i> The network user should be authenticated by means of a client certificate.
SS1-12.0	<i>Secure authentication of service user</i> The service user should be authenticated by means of a client certificate.
SS1-13.0	<i>Single sign-on for network and services</i> A user should only have to authenticate once for network and service access.
SS1-14.0	<i>Generation of event based service records</i> 3 rd party services should generate charging records for specific usage events.
SS1-15.0	<i>Flexible mediation</i> The rating facility must handle data sent to it by different mediation components.
SS1-16.0	<i>Flexible, correct and comprehensible billing</i> Billing should be efficient, correct and comprehensible. Charging schemes should allow for consistent rating.
SS1-17.0	<i>Provision of user accounts</i> The portal should support administration of user account, i.e. create, change, delete, etc.
SS1-18.0	<i>User Preferences</i> e.g. Personalised billing
SS1-19.0	<i>Generation of network usage record on disconnection from network</i> On disconnection from the wireless network a record should be sent to the IPDR store detailing the time during which the user was connected.
SS1-20.0	<i>Usage of network location services</i> A third party application must be able to obtain a user's location from the network

SS1-21.0	<i>Use of standardised performance and service quality indicators</i> The service provider must be able to use standardised indicators to measure and specify the service quality and network performance.
SS1-22.0	<i>Definition of network resources and configuration of end-to-end network architecture</i> The resources used to construct a service and the provider domains from which they are drawn must be well defined. A network architecture to support end-user services from the mobile terminal to the serving computer must be specified and boundaries of provider domains within this network architecture must be clearly demarcated. Wherever possible (e.g., in case of UMTS/GPRS access networks), configuration of resources and provider domains must conform to internationally agreed network architecture.

The table below shows a mapping between the sub-trial requirements and the OSS components implementing these requirements:

Req. ID	Restaurant Service	Portal Access Manager	CSC Manager	Billing Engine	AA Manager	Accounting Repository	RBS	FMA	Q3ADE IPDR Store	SLA Manager	Report Log	MAC	Metering Component	NPE
SS1-1.0		●	●		●									
SS1-2.0		●	●											
SS1-3.0	●				●									
SS1-4.0										Note1		●		
SS1-5.0														●
SS1-6.0											●	●		
SS1-7.0								●	●					
SS1-8.0								●	●					
SS1-9.0	●			●			●	●	●					
SS1-10.0								●	●					
SS1-11.0													●	
SS1-12.0	●				●									
SS1-13.0	●												●	
SS1-14.0	●							●	●					
SS1-15.0							●							
SS1-16.0				●			●							
SS1-17.0		●												
SS1-18.0				●		●	●							
SS1-19.0													●	
SS1-20.0	●	●												
SS1-21.0												●		●
SS1-22.0												●		●

Note 1: Mobile Portal Service Provider must be able to analyse product KQI values achieved against the product KQI objectives in the SLA and issue warnings and alarms if the delivered QoS does not meet the specified QoS.

3.3 Functional Architecture

The OSS functions evaluated in this section are fully described in [D8].

The architecture is evaluated from the perspective of the following stakeholders:

- Customer
- Mobile Portal Service Provider
- 3rd Party Service Provider

3.3.1 Customer

This subsection describes architecture evaluation results from the perspective of a Customer. Customers have a subscription contract with the Mobile Portal Service Provider, which provides them with the Restaurant and Smart Home services bundled in a single service product. Customers receive an itemised bill for the services they use.

The evaluation results of OSS functions that relevant to the Customer are as follows:

Authentication and Authorisation

The customer enters his login name and password to enter the portal. This completes the authentication process from the view of a customer. If there is a session timeout, the customer has to log in once again. The authorisation occurs transparently to the customer. As an example he can only see those menu points that he is authorised to see.

Billing

The main purpose of this function is to produce an itemised invoice bill based on the user's services consumption. It should be noted that this function only provides a "query and transformation of results" functionality for previously rated usage records based on the NDM-U IPDR specification. The component implementing this function, the Billing Service, queries the IPDR Store/FMA for the relevant IPDR document and produces a customised bill based on the customer's preferences. From the customer's point of view, flexible, personalised billing is a crucial part of accounting. However, billing is only near real-time as this interface relies on the successful delivery and rating of usage records. The Billing function should also serve as a customer-facing interface where the customer can choose to get a bill based on all his records or a selection based on particular parameters such as service or date period.

3.3.2 Mobile Portal Service Provider

Mobile Portal Service Provider carries out tasks of subscription management, maintenance of QoS, and billing functionality associated not only with the Mobile Portal service but also with the Restaurant and Smart Home services.

The evaluation results of OSS functions that are relevant to the Mobile Portal Service Provider are as follows:

Authentication and Authorisation

The customer sends his login name and password through the network. The login name and password are stored temporarily at the Mobile Portal Service Provider side for the session. The authorisation is dependent of what is desired by the customer. For service usage purposes the QoS information of the services comes into the authorisation process. For authorisations aiming at customising the user interface, such as show/hide menu points, no additional information is needed.

Billing

The Billing function prepares an account of a defined customer to be invoiced with the relevant service usage information. This information can serve to make adjustments to the charges (Rating component) to either attract more customers or increase profit.

Service Quality Management

The main purpose of this function is to ensure that the overall quality of service delivered to customers meets the service KQI objectives. Service KQI values are derived from KPI values. The component implementing this function, MAC, fulfils this function. However, it must be noted that establishing a clear-cut mapping between KPIs and KQIs for end-to-end service quality and developing a transformation algorithm is a complex task. Nonetheless, algorithms for measuring service performance were developed in order to ascertain the QoS being obtained by end users in a mobile environment.

Resource Performance Management

The main purpose of this function is to ensure that the overall performance of the resources involved in delivering services to customers meets the resource KPI objectives. The component implementing this function, NPE, fulfils this function with some limitations. The limitations arise because it is difficult to measure performance of resources that reside in several administrative domains. Some of the domains can be accessed, some can be accessed with restrictions, and some cannot be accessed at all. As a result, the measurement does not give a complete picture of the performance of all resources involved and, consequently, service quality cannot be ascertained with a great accuracy.

Rating

The purpose of this function is to convert the measurement of service specific usage data into realistic end-charges for the customer of the service. It should be noted that this function only provides a cost allocation for each reported activity of the overall service that has been consumed; invoice preparation is a function of Billing. Usage data can include parameters that record volume, content, event and quality type information about the service being consumed. The rules and tariffs for rating a service (which are encapsulated in a charging scheme) are applied to the generated usage data producing a rated record of the service usage. The component implementing this function, RBS, fulfils this function completely.

Inter-provider Settlement

The purpose of this function is to determine how the revenue generated from the service usage should be distributed among the collaborators (MNO and 3PSP) in the service. The

Billing Engine demonstrates how this can be achieved by producing a separate invoice detailing the proportion of the service charge that should be passed on to the 3PSP by the MNO.

Federated Usage Mediation

The purpose of this function is to retrieve proprietary usage data from service metering functions and to mediate these into standardised IPDR record format. After mediation, the IPDR records are pushed as IPDR Documents to a rating function. When session information is available in the usage records, those IPDR records originating from the same service and pertaining to the same session are aggregated into the same IPDR Document before being forwarded to the rating functions. Two similar mediation components were developed in this sub-trial, both fulfilling the requirements for flexible federated usage mediation. The two mediation components were used to demonstrate and evaluate how mediation services can be offered as distinct service components, thus allowing a service provider to switch among mediation service with a minimum of integration effort.

3.3.3 Third Party Service Provider

3PSPs have contracts with the Mobile Portal Service Provider. They provide their services via the portal of Mobile Portal Service Provider. The bill for service usage is sent to the Mobile Portal Service Provider.

The evaluation results of OSS functions that are relevant to the 3PSP are as follows:

Generation of charging records

End-users are charged for using different features of the third party service. Events that are charged for are for example a table reservation in the Restaurant Service, or switching on or off a light in the Smart Home Environment. The trial demonstrates the generation of proprietary accounting records (PARs) by these third party services, which are sent to the *Federated Usage Mediation* component for further processing. Both event-based and session-based accounting for third party services is demonstrated.

Different mediation components can be used, for example depending on the home network provider of the user. This is demonstrated by implementing 2 different mediators, which are both used by the same third party service.

Inter-provider Settlement

Obviously it is in the interest of the 3PSP to share the revenues that are generated by end-users using their services. As described in the previous section, this is demonstrated by producing an invoice that details the proportion of the service charge that should be passed on to the 3PSP.

Authentication & authorisation

Proper authentication of end-users is an important aspect of 3PSP services in a 3G environment, for multiple reasons:

- services can be personalised,
- end-user privacy can be ensured,
- services can be enhanced by using 3G network capabilities (e.g. user location),
- services can be charged for,
- multiple authorisation levels can be offered,

- etc.

The trial demonstrates two ways to perform authentication & authorisation:

- *Portal-based*: the user logs into the portal of the Mobile Portal Service Provider, before accessing the third party service. The 3PSP and the Mobile Portal Service Provider exchange authentication and authorisation information when the user accesses the third party service. The greatest benefit for the user is that he/she only needs to login once for accessing all registered services (“single sign-on”).
- *Certificate based*: the user installs a X.509 client certificate on the terminal device. When a third party service is accessed, authentication is performed automatically between the user’s web browser and third party web server. In the trial, the actual authentication is performed by the Mobile Portal Service Provider, making it possible to use one client certificate for both network and service access (“zero sign-on”). Although this approach requires some capabilities on the terminal, it is very convenient for the end-user, without jeopardising authentication security.

3.4 SID Model

The SID Domains and information entities evaluated in this section are fully described in [D8].

The evaluation results of the information entities that are relevant to the Mobile Portal Service Provider are as follows.

SID Domain	SLA
Which entities are used?	ServiceLevelSpecifcation ServiceLevelSpecConsequences, ServiceLevelSpecObjectives, ServiceLevelSpecParameter, KeyQualityIndicatorSLSParam, KeyPerformanceIndicator
Which relationships are used?	<i>aggregation</i> <<unmetObjectiveResults>> between ServiceLevelSpecifcation and ServiceLevelSpecConsequences <i>aggregation</i> <<intendsToMeet>> between ServiceLevelSpecifcation and ServiceLevelSpecObjectives <i>association</i> <<expressedBy>> between ServiceLevelSpecObjectives and ServiceLevelSpecParameter <i>generalization</i> between ServiceLevelSpecParameter and KeyQualityIndicatorSLSParam <i>dependency</i> between KeyQualityIndicatorSLSParam and Availability, Accuracy and ServiceTransactionSpeed
Which attributes were added?	AggregatedServiceAvailability, AggregatedServiceAccuracy, aggregatedServiceTransactionSpeed, manAvailability, ednAvailability, canAvailability (here <i>man</i> stands for mobile access network, <i>edn</i> for external data network, and <i>can</i> for content/application network.) up-time, down-time, httpRTD, jitter, SDUCCount, BitErrorRate, TimeToRegister.
Which entities/relation were not used?	AggregatedServiceAccuracy, Accuracy AggregatedServiceTransactionSpeed, ServiceTransactionSpeed

Justification for not using these entities	The mapping between KPIs and <i>Accuracy</i> and <i>ServiceTransactionSpeed</i> KQIs is not sufficiently clear and needs further research.
Which entities/relations were added?	Entities: Availability, Accuracy, ServiceTransactionSpeed Relationships: <i>dependency</i> of AggregatedServiceAvailability on Availability <i>dependency</i> of AggregatedServiceAccuracy on Accuracy <i>dependency</i> of AggregatedServiceTransactionSpeed on ServiceTransactionSpeed
Justification for adding these entities.	To measure availability in the mobile access network (ie, UMTS core network), external data network (ie, Internet), and in the content network (ie, SP's intranet) and to aggregate these three availability values in a single KQI value. Values of accuracy and transaction speed are also measured according to relevant factors and aggregated in a single KQI value.
Should these entities be added to the SID model?	Yes.

SID Domain	Billing
Which entities are used?	IPDRDoc, Invoice, Customer, BillingAddress, BillingPreferenceSet, BillingPreference, PAR,
Which relationships are used?	Uses, hasIPDRs, generates PAR – IPDR, IPD – IPDRDoc
Which attributes were added?	
Which entities/relation were not used?	None relevant to billing functions
Justification for not using these entities.	-
Which entities/relation were added?	None.
Justification for adding these entities.	-
Should entities be added to the SID model?	-

3.5 OSS Components

3.5.1 Portal Access Manager

A functional description of this component can be found in [D8, section 6.1.2.5].

The Portal Access Manager does not include any logic and is responsible for drawing up the user interface by calling operations of other components that provide the needed logic.

3.5.2 CSC Manager

A functional description of this component can be found in [D8, section 6.1.2.2].

The CSC Manager provides the database contents in the form of methods including the needed logic by other components. Those methods include e.g. the management of parties, services, products and SLA based data like the service level specification information

provided for the SLA Manager. The standards used for the development include EJB 2.0, JBoss 3.0.4, Java 1.4.

3.5.3 Billing Engine

The Billing Engine produces a detailed record of billable usage activity for a particular service in a convenient customer invoice. The Billing Engine is also described in AlbatrOSS D8 section 6.1.2.10 and D9 section 4.8.3.3.

The technologies used for the implementation of Billing Engine include XSLT, XPath, Apache Formatting Objects Processor (FOP), .NET Web Service (C#).

The standards used for the development include NDM-U IPDR version 3.1.1 [IPDR311], SOAP/WSDL, XPath 1.0, and XSLT 1.0.

In AlbatrOSS Trial Phase 1, the Billing service generated invoice bills based on the last usage records stored in the accounting repository. Each service produced a clear table representing the customer's aggregated network and service consumption.

In Trial Phase 2 a different approach was adopted. Customer parameters such as service type and date period were used to query for the relevant record. Also, due to new requirements both customer bills and 3PSP settlement invoice can now be produced.

3.5.4 AA Manager

A functional description of this component can be found in [D8, section 6.1.2.1].

The main purpose of the AA Manager is to provide the JAAS API (Java Authentication and Authorization Service), and to fully implement the JAAS framework in a way, so that the AA Manager obtains all information needed during the authentication and authorisation processes by calling the EJBs of the CSC Manager. The standards used for the development include EJB 2.0, JBoss 3.0.4, Java 1.4 with integrated JAAS 1.0.

3.5.5 Accounting Repository

The Accounting Repository component is a native XML database providing storage and retrieval services to the accounting related components. The Accounting Repository is also described in AlbatrOSS D8 section 6.1.2.9 and D9 section 4.8.3.4.

The Apache Xindice project was used as the implementation of the Accounting Repository. The decision to adopt a native XML database over traditional object or relational implementations was based on the Accounting information model, the IPDR NDM-U format. Where traditional information management techniques attempt to map the data contained in XML documents to some internal representation such as a table or object, native XML databases retain the integrity of the data by storing documents in their existing format. The XML document is treated as a discrete unit of data and can be queried for individual element data or complete structures existing inside the document.

The standards used for the development include XPath 1.0, XML:DB [XML DB], and XML-RPC [XML RPC].

3.5.6 RBS

The RBS is responsible for rating usage data originating from the services supported in this sub-trial (Restaurant Service and Smart Home). Usage records can be supplied by the FOKUS Federated Mediation Adapter or UHC Q3ADE IPDR Store components.

The RBS is also described in section 6.1.2.8 of [D8] and section 4.8.3.2 of [D9].

RBS is implemented using Visual Studio .Net, COM+, C#. The standards used for the development of RBS include NDM-U IPDR version 3.1.1 [IPDR311], SOAP 1.1 /WSDL 1.1.

In AlbatrOSS Trial Phase 1 service usage records were developed in accordance with the IPDR NDM-U specification, which advocated the usage of XSLT schema to document record structure. Each service or network provider produced a schema specification detailing the element and attribute content that captured the usage data produced from their service. The only constraint placed on the network or service provider was to include the required element structure of the IPDR specification, all other content was provider specific.

In Trial Phase 2 a different approach was adopted. A single record structure was used to capture both network and service usage data. Specific elements were introduced by the PAR format to indicate the nature of the usage record. An event element indicated the presence of a service event such as operating the light in the Smart Home or requesting restaurant location information through the Restaurant Service. It could also contain a duration element to indicate the presence of metered usage data such as length of a UMTS or WLAN session.

Both approaches are valid and proved successful in the context of the sub-trials they were applied. The network/service specific usage record structure required more integration effort between the individual network and service providers and the mediation and billing functions. However, the generic usage record structure also required significant integration effort as usage data produced from the network or service level had to be mapped to the structure of the PAR record format for presentation to the components fulfilling the Billing function.

3.5.7 FMA

This Federated Mediation Adapter (FMA) is described in section 6.1.2.3 of [D8] and in section 4.2.3.3 of [D9].

The implementation technologies used for the development are Java 1.4.1 and JBoss 3.0.4, a J2EE compliant application server with Apache Tomcat web server. The NDM-U IPDR version 3.1.1 [IPDR311] specification is used for the development of FMA.

The FMA is one of the two mediation components used for the usage mediation, the other one being Q3ADE IPDR Store (section 3.5.8). Please refer to the conclusion of this chapter (section 3.8) for a general evaluation of the federated mediation concept.

3.5.8 Q3ADE IPDR Store

This component is described in section 6.1.2.11 of [D8] and in section 4.7.3 of [D9].

The Q3ADE IPDR Store is implemented entirely using scripting code, which is executed on the Q3ADE platform, except for persistency of IPDR records, which is provided by an SQL database (MySQL). The scripting language used is UH Communications proprietary LMIF language with LHP extensions. The standards used for the development include NDM-U IPDR version 3.1.1 [IPDR311], SOAP 1.1/WSDL 1.1.

The Q3ADE IPDR Store is one of the two mediation components used for the usage mediation, the other one being FMA. Please refer to the conclusion of this chapter (section 3.8) for a general evaluation of the federated mediation concept.

3.5.9 SLA Manager

The SLA Manager is an additional component needed to fulfil the sub-trial requirements for Service Quality Management. It is not part of sub-trial architecture described in [D8] and [D9].

The component is responsible for maintaining customer SLAs and for ensuring that the delivered QoS meet the QoS specified in customer contracts or other product specifications. It ensures that the provisions in the contract are being met and issues warnings and alarms if this is not the case. This means that it must be able to monitor, analyse and report on the actual QoS achieved compared with the QoS specified in the SLA.

The SLA Manager is implemented using JBoss version 3.0.4. The functionality is based on concepts laid down in SID [SID-1A] model and eTOM business process framework [GB921].

3.5.10 Report Log

A description of this component can be found in section 4.2.3.8 of [D9].

JBoss version 3.0.4 is used for the development of Report Log. The work is loosely based on guidelines laid down by eTOM business process framework [GB921].

Report Log is a convenient tool for the service provider who may want to browse through a log of service quality alarms.

3.5.11 MAC

This component is described in section 6.1.2.4 of [D8] and in section 4.2.3.4 of [D9].

MAC is implementation using Java version 1.4.1 and JBoss version 3.0.4. The development is based on SID [SID-1A], [GB921], [ITU-T Y.1541], [ITU-T I.380], [3GPP 23.107], [3GPP 23.207].

As an additional evaluation result it is worth mentioning that MAC can be tested as a unit, as well as in conjunction with SLA Manager. In the latter case, MAC and SLA Manager monitor SLA violations. For example, if the objective for overall service availability is set to 90%, then MAC should be able to report the SLA Manager if service availability is below 90%. It can be concluded that standards have greatly helped in the design and development of MAC and configuration of end-to-end architecture on the UMTS test bed. The WSMH provides a useful framework of performance and service quality indicators.

3.5.12 Metering Component

Please see section 4.3 of [D9] for description of this component.

Implemented on Linux 2.4.23 with hostAP extensions, using Java 1.4.1, Jakarta Tomcat, and Python 2.2/2.3.

In evaluation we found that the standard 'ping' implementation could not be relied upon to determine network presence. This is because some hosts have this facility turned off. Instead we found that the 'arping' facility was pretty much universally supported.

3.5.13 NPE

This component is described in section 4.2.3.10 of [D9].

NPE is implementation using GSOAP and C++. The development is based on [RFC 2330], [RFC 2681], [ITU-T Y.1541], [ITU-T I.380], [3GPP 23.107], [3GPP 23.207].

As an additional evaluation result it is worth mentioning that ICMP and its PING implementation are efficient ways of reporting status of remote resource. For the performance probes, on which NPE relies, PING implementation is used.

3.5.14 Restaurant Service

A description of this component can be found in section 4.4.3.2 of [D9]. Although the Restaurant Service (RS) is a demonstration service and not an OSS component, it is relevant since it participates in some of the OSS functions (e.g. generation of usage records).

Relevant standards: Java, J2EE (JBoss), JAAS, WSDL/SOAP/XML(Axis), HTML/HTTP (Tomcat), X.509.

This evaluation focuses on the aspects of the service relevant to OSS functions:

User Authentication

For user authentication the RS uses X.509 client certificates. The implications are that a client certificate must be distributed to each client before the service can be used. An implementation specific consequence is that HTTPS must be used to access the service.

The use of client certificates for service authentication provides automatic zero sign-on capabilities: a typical HTML browser (e.g. Internet Explorer) will automatically respond to a web server's request for a client certificate, as it looks for a certificate that matches the server certificate presented. Provided there is exactly one matching certificate (which depends on the deployment scenario and business models used) there is no need for user interaction: the end-user notices no difference.

There is a performance penalty due to the use of HTTPS (an implementation restriction imposed by Tomcat), which makes this approach less suitable for resource limited devices such as 3G handsets. In addition, client certificates are still not very well supported on such devices.

On the server side, implementations for using client certificates exist but do not seem to be commonly used. We have had to modify the web server code slightly to be able to use them.

At a business level, the question is whether the deployment cost of client certificates (often quoted as a reason not to use client certificates) is real or imaginative. In AlbatrOSS we did not address large-scale deployment, although our feeling is that it is possible to build a secure distribution system at low cost.

Usage records

For the generation of usage records we adhered to the (proprietary) interface defined within the project. A relatively simple WSDL specification was used to describe the server side functionality, and it was straightforward to generate matching client side code using Axis. The flexibility of this approach was demonstrated by allowing per-session specification of the PAR server address: the service supports simultaneous users from different providers, generating PAR records to different servers. In the prototype case, this means that records are sent either to Germany (FOKUS) or Denmark (UHC). Again, the user notices no differences.

Overall, the Restaurant Service served well as an exemplary 3G application service, and contained many aspects relevant to a 3G OSS. It was developed as a stand-alone service, with clearly defined interfaces to other components developed by both Lucent Technologies and other partners within AlbatrOSS. All interfaces use a uniform approach (WSDL) and are clearly separated between OSS functional boundaries, which suggests it would be possible to standardise OSS functionality using this technology.

3.6 Implementation Technologies and Standards

3.6.1 Technologies

3.6.1.1 Java

Java is becoming such a standard technology that it perhaps does not even need to be mentioned in this section. The main benefits in an OSS context are the promise of develop once – deploy everywhere, which we found to be true: all Java based components can be run on different hardware / operating system platforms by simply copying the contents of the directories that contain them.

A potential drawback of Java is performance and resource related: on average Java programs tend to use significantly more memory and run slightly slower than components written in assembled languages such as C++. For prototypes and research projects this is hardly an issue, but large-scale deployment requires more investigation and careful optimisation.

3.6.1.2 J2EE/JBoss

Several partners within AlbatrOSS used JBoss (version 3.x) which is an open source implementation of the J2EE reference architecture developed by Sun Microsystems [J2EE].

MAC, FMA, CSC Manager, Portal Access Manager, Report Log and AA Manager use JBoss as an application server and JDBC for database access. An advantage of JBoss is that it is open source software and publicly available.

3.6.1.3 LMIF

The scripting language LMIF language with LHP extensions is used for the development of Q3ADE IPDR Store. It allows scripting code to be executed as active server pages on the Q3ADE platform web-server.

The LMIF scripting approach performed well in the trials, and offered a high degree of flexibility, e.g. it allows that the IPDR Store functionality can be modified even when the service is running, to correct errors or adapt to changes in the exchanged usage record formats.

3.6.1.4 Apache Formatting Objects Processor (FOP)

For a preferred format being PDF, etc. the construction is a two-phase process:

Using XSLT, an XML Formatting Object document (or .fo) is produced from the resulting usage records.

This .fo document is processed and transformed to one of the formats supported by the FO Processor (PDF, PS, PCL, SVG, Print, AWT, MIF, or TXT). This facility required the installation of the Visual J# .Net Redistributable Package to handle the Java based FO Processor. The Billing Engine is developed using XSLT, XPath, Apache Formatting Objects Processor (FOP), .Net Web Service (C#).

3.6.1.5 .Net and C#

The Rating Bureau Service (RBS) component was implemented entirely in COM+ and deployed as a COM+ application on a dedicated machine. The functionality of the RBS was exposed through web service interfaces implemented in Microsoft's C# language. The Visual Studio .Net development platform provides various developer tools and functions for creating and deploying web services with the focus on abstracting the complexities of the underlying XML based standards. However, some appreciation of the SOAP, UDDI, WSDL and the more recent WS Security specifications is required to successfully build an interoperable web service.

3.6.1.6 Accounting Repository

The Accounting Repository is built on Java SDK 1.4, Apache Tomcat 4.1 (Web application server) and Apache AXIS 1.1. It uses the Apache Xindice Native XML database implementation to store rated and unrated usage records passed to the RBS from the mediation layer. Apache Xindice is a relatively immature implementation of an XML database when compared to the other implementations (commercial and non-commercial) available today. It provides an implementation of the XPath 1.0 specification for document and collection traversal. However, the XQuery specification has now become the de-facto standard for querying among most other implementations.

3.6.2 Standards and Recommendations

3.6.2.1 IPDR NDM-U

IPDR Specification NDM-U [IPDR311] is used for the development of Billing components, i.e., Q3ADE IPDR Store, FMA, RBS, Accounting Repository and Billing Engine.

For the development of Q3ADE IPDR Store, the NDM-U specification has proven to be a flexible standard, and has accommodated the requirements for the sub-trial 1. In the trial, the IPDR format was extended with a proprietary extension to structure IPDRs for the rating and billing service. The trial has used the XML-based coding of IPDRs for the mediation services, which has been sufficient for this trial. It is worth considering the binary XDR format for high volume processing.

For the development of RBS, NDM-U IPDR version 3.1.1 has been used as the standard covering the IPDR document format and mode of document transfer between mediation and rating components. This standard has also been used for the development of Billing Engine where it has been useful for describing the service usage in a standard way.

As a side note, the flexibility and generality of this standard is at the same time its point of weakness: Whereas the IPDR standard defines the transfer protocol and subscription mechanisms, it is currently a very open standard with respect to the structure of the IPDR Records, except for general rules on the use of time formats. From a high level point of view it only dictates that every IPDR record starts with “<ipdr>” and ends with “</ipdr>”, everything else is optional or extensible by proprietary mechanisms. The danger here is that all companies do this (like we have) resulting in a jungle of too many semi-standards. The usefulness of the IPDR standard should increase as more service specific standards are added to the standard.

3.6.2.2 TM Forum Recommendations and Guidelines

The OSS functions and information model that supported the sub-trial are derived from TMForum framework, most notably eTOM [GB921], and concepts laid down in SID model guidebooks [SID-Concept] [SID-4SO], [SID-1A].

The OSS functions used by MAC and NPE are based on the Service Quality Management and Resource Performance Management processes of the eTOM Assurance business process [GB921].

SID models have provided a set of entities that can be used for the development of information model. The information model required for MAC and NPE adopts those entities, proposed new entities and attributes reflecting emerging requirement for end-to-end QoS measurement. Within AlbatrOSS we added IPDR elements to our SID model.

Concerning end-to-end aspect of service quality, there are three main views that are important for analysis: **network**, **service**, and **user/customer**. A hierarchical relationship between these three levels is derived from [GB923] and used in the development of NPE and MAC. The hierarchical relationship reflects the customer scenario used in this sub-trial whereby Restaurant service and mobile network service are combined in a service product at the user/customer and service quality objective is set at this level. Service quality, which resides

at the service level, is derived from the value of resource performance indicators, which resides at the network level.

3.6.2.3 3GPP Technical Specifications

There are two Technical Specifications [3GPP 23.107] and [3GPP 23.207] that proved to be very useful for the definition of end-to-end network architecture, QoS classes and QoS measurement scenario. In fact, the *Interactive* class of service (e.g., a Web-based service) undertaken by QoS scenario is defined in [3GPP 23.107].

3.6.2.4 W3C Recommendations and Specifications

W3C Recommendations are used to specify information entities such as usage records and charge records. To assist the development and integration work, they are also used for the definition of component interfaces and protocol specification.

SOAP/WSDL

SOAP/WSDL is used extensively within AlbatrOSS. The “Simple Object Access Protocol” (a misleading name) is a technology that seems to be self-fulfilling, as the “simplicity” of use comes from generally available tool support, which is due to its popularity as a result of the Web Services hype. As such, this broad support is arguably its best feature.

Within the context of AlbatrOSS we found that SOAP (and its companion WSDL) is generally straightforward to use, provided you use the same tool vendor and tool version for both sides (client and server). Interoperability is still a major issue, and requires in depth knowledge of the more obscure aspects of the standard.

The ease with which new interfaces can be defined (in WSDL) has a drawback, namely the bloat of available interface definitions. For example, for communication of service usage details we opted to define our own interface, loosely based on the IPDR standard. We probably could have used an existing standard as well, simply by adapting our own interpretation of semantics of certain parameters. The danger here is that everyone takes this approach, resulting in a web services mess.

We did find that one of the qualities for which SOAP is praised is true: It is easy to traverse firewalls. On the other hand, once SOAP becomes widely deployed this will likely change.

SOAP/WSDL is used for the development of Q3ADE IPDR Store. Even though the SOAP protocol standard is comprehensive and allows for precise specification of service interfaces using WSDL and XSD, it is our experience that a high level of expertise is required to fully develop and integrate SOAP based interfaces. Discrepancies in how the service specific specifications are interpreted e.g. through code generation, still complicate the process of generating and integrating a SOAP based interface.

SOAP 1.1/WSDL 1.1 specifications are also used for the development of RBS, Billing Engine and MAC. They cover messaging structure and client-neutral service definition and allow components written in traditionally incompatible languages to interoperate easily using a common messaging protocol.

XPath

XPath 1.0 is used for the development of Accounting Repository and Billing Engine. In case of Accounting Repository, it provides an implementation of the W3C XML Path language specification for document (and document fragment) retrieval. In case of Billing Engine, W3C XML Path language specification is required to navigate through the Accounting Repository and retrieve the relevant usage records.

XML:DB

XML:DB [XML DB] is used for Accounting Repository. XML:DB API is being developed by the XML:DB Initiative to facilitate the development of applications that function with minimal change on more than one XML database. This is similar to the functionality provided by JDBC or ODBC for providing access to relational databases. Xindice implements Core Level 1 of the XML:DB specification.

XML-RPC

XML-RPC [XML RPC] is used by the accounting components to invoke methods (add, retrieve, delete, index document etc.) on the accounting repository. XML-RPC is a simple and lightweight remote procedure calling specification that utilises HTTP as its transport protocol and XML as its encoding mechanism. It maintains a relatively simple type system comprising of the standard basic types such as string, int, boolean, double, dateTime and base64 and complex types like struct and array. Using this type system complex data structures can be constructed using only the essential elements. XML-RPC has also been cited as the precursor to the SOAP messaging protocol but with much fewer complexities.

XSLT

XSLT 1.0 is the W3C XML document transformation specification and used for the development of Billing Engine. It is used to enforce a structure on the usage records retrieved from the Q3ADE IPDR Store/FMA.

3.6.2.5 IETF RFCs

IETF RFCs developed by IPPM Working Group [RFC 2330] have been useful in deriving KPI definitions.

The RTD (Round Trip Delay) KPI is derived from the RTT definition given in [RFC 2681], which defines a metric for round-trip delay across Internet paths. The SDU Loss KPI is derived from One-way Packet Loss Metric definition given in [RFC 2680]. The KPIs reporting capacity of network and services, and throughput KPI are derived from [RFC 3148], which proposes a framework for defining empirical bulk transfer capacity metrics.

The ICMP protocol [RFC 792] is used mainly to report the values of two main KPIs, up-time and down-time, of a remote server. The technique that is used to develop these KPIs made use of PING implementation of ICMP protocol.

3.6.2.6 ITU-T Recommendations

ITU-T Recommendations are used mainly for the development of MAC and NPE.

ITU-T Recommendation [ITU-T I.380] provides information on where the performance measurement points must be placed in IP network. It also provides information on IP service availability. It is used for the development of NPE and KQIs. Another Recommendation [ITU-T Y.1541] provides general requirements for network performance for IP-based services, which are useful for developing MAC and NPE.

3.7 Deviations

The Mobile MeetMe demonstration service described in [D7] and [D8] was replaced by the Restaurant Service. This third party service demonstrates all OSS concepts that are part of the trial, but has a simpler structure than the MeetMe service, both for the end-user and the service developer. It is therefore more suitable in a prototyping environment.

The SLA Manager described in section 3.5.9 was added to the architecture to fulfil the Service Quality Measurement requirements.

3.8 Conclusions

The main aims of this sub-trial were to (1) set up necessary software and hardware infrastructure, including configuration of UMTS, WLAN, GPRS and IP network test-beds, (2) assemble OSS components into an integrated sub-trial system, (3) operate the integrated sub-trial system to manage provisioning, QoS and charging of 3G service and networks, and finally (4) evaluate the integrated system by a set of criteria.

The 3G services used for testing the OSS components are Restaurant and Smart Home services. The main conclusion that can be drawn from the first three points mentioned just above is that an integrated sub-trial system can manage provisioning and charging of 3G services. Quality of services provided on UMTS and IP test-bed can also be managed.

Sub-trial evaluation concludes that the integrated sub-trial system fulfils the requirements that were captured in the phase two of the project and conforms to concepts and guidelines laid down by international standard bodies. Guidelines such as eTOM, SID model, and IPDR specification have continued to be useful for the design and implementation of OSS components. They have helped greatly in maintaining a common understanding of OSS functions and information entities required, and an overall picture of the integrated OSS.

Development and integration technologies such as SOAP and WSDL proved their usefulness when diverse set of OSS components were integrated in a single OSS, which was distributed over test-beds in Ireland, Germany, Denmark, The Netherlands, and Spain.

As an overall conclusion of QoS scenario of the sub-trial, it can be said that finding out right performance metrics and mapping them onto right service quality indicator is a crucial task. Developing algorithms that transform performance metrics into service quality value is equally crucial. More compound algorithms and detailed mapping will be considered for future research.

This sub-trial successfully demonstrated that components MAC and Q3ADE IPDR Store are loosely coupled to RBS, thus proving an important principle that components should be loosely coupled. The providers of Restaurant and Smart Home can select a mediation device

(Q3ADE IPDR Store and FMA) they want to use. The RBS can use either of the two of devices for receiving IPDR documents. Therefore it proves that interface between RBS and mediation devices is interoperable and mediation devices are loosely coupled to RBS.

The accounting components provided for this sub-trial demonstrated that billing and rating need not necessarily be tied to particular services and that once a service can be described in terms of its usage and metrics any conceivable service can be charged for using similar means.

This sub-trial has demonstrated the necessity of federated mediation of billing information. As all services in the second phase trials migrated to SOAP based interfaces, the project decided to use one common format (PAR) for all usage records generated from the trial services. Even though this cannot be expected in a real world scenario, this approach was applied in order to keep trial integration effort low, keeping in mind that the first AlbatrOSS trials had already demonstrated successful mediation of various proprietary usage records formats.

The common XML PAR structure was developed primarily to accommodate the requirements of end-user services used in the trial, but it is our experience from the trial that it is more difficult to accommodate requirements from both services and rating functions in a common format.

In contrast to the AlbatrOSS trial 1 phase, where the mediation function was responsible for performing time-based correlation of usage records across services, the second phase trials were restricted to perform correlation of records within each service based on a session ID. Whereas the time-based correlation proved difficult to implement, the session-based correlation was uncomplicated and straightforward to implement.

The billing functions in the trial, spanning from metering, over mediation to rating, was executed as geographically distributed components (service metering (Spain/Holland), mediation (Germany/Denmark), rating (Ireland)). The highly distributed architecture of the trial proved uncomplicated and worked reliably for the trial.

The trial did not consider aspects of security in relation to mediation. This is for future research.

4 Evaluation results: Inter-technology and Inter-domain Roaming

In the second AlbatrOSS trial system, the *Inter-technology and Inter-domain Roaming* sub-trial demonstrates an enhanced version of the *Prototype Roaming* sub-trial from the first trial system. The key aspects of this trial are:

- Roaming contract management
- Roaming authentication and authorisation
- Billing and cost information
- Best connection
- Value-added services
- Discriminative seamless roaming

The component architecture and interfaces of the sub-trial are explained in [D9, section 5.2]. Information about the business context for this sub-trial and the functional architecture can be found in [D7, section 4.3] and [D8] respectively.

4.1 Business processes support

Roaming Contract Management for MNO

The roaming contract management process should be designed in such a way that incumbent operators can cooperate with a big number of smaller WLAN providers in an efficient way. Processes like contract negotiation and contract change management should be automated and supported.

The implemented Contract Manager offers a web interface enabling the MNO to *create/change roaming contracts* with WLAN-Provider in an easy way. The created roaming contract can be activated automatically. Updates/changes of a roaming contract can be carried out “online” and affecting roaming service directly. The content of the roaming contract is kept simple because the main focus of the sub-trial is to demonstrate how roaming contracts with WLAN providers can be managed efficiently, but not to demonstrate how a “real” roaming contract can be implemented. *Contract implementation process* is not implemented with the whole details described in the D7.

Authentication and authorisation of roaming customers

The authentication and authorisation (AA) mechanism in 2.5/3G networks is based on SIM or USIM. In public and private WLANs other mechanisms are typically used, for example web-based login, shared secret, MAC address filtering, etc. To guarantee roaming between 3G and WLAN network, there is the need of integration of WLAN AA approach into 3G networks.

In the sub-trial, the WLAN AA mechanism is based on the IEEE 802.1x access control framework, which combines a number of authentication protocols and encryption mechanisms. An AAA server deployed by the MNO performs the actual authentication of roaming users. The mechanism makes use of X.509 digital certificates, which are generated and distributed by the MNO.

Billing and cost information in real time

Online cost & billing information services are established part in 2G OSS but focusing on the “after-use” information for customer, just to know “how much do I have to pay?”. In the access-diverse landscape of 3G, customers want this information to make the decision of “what is the cheapest access for?”. In reality, operators are not really willing to have

too much transparency, especially regarding the cost. But the more complex roaming market, the heavier the press on MNO, above all, from the side of telecommunication regulation and customer protection organisations drives the development towards this transparency.

The Network Roaming Manager supports the process of offering billing and cost information in real time in this sub-trial. The Network Roaming Manager offers customer who roams in WLAN hotspot the information about the tariff, cost in real time. This information is shown on the customer terminal and updated whenever the customer changes the access networks.

Best connection

Another problematic aspect in the access-diverse 3G environment is how to support a customer choosing the best access for her/his purpose? The interpretation of “best” can be very different from one situation to the next, and depends on the customer’s momentary needs: “best cost”, “best bandwidth”, “best quality”, etc. The “best” for an operator is if customers use its networks but not the networks of the competitors. However, operators want to offer the “Best Connection” choice to the service consumer as a value added service.

The decision process of what is the best connection has been designed in the sub-trial based on a combination of customer preferences (e.g. cheapest connection, best bandwidth, etc.) and current access environment (e.g. tariff, network capabilities like bandwidth, QoS etc.).

Value added services – Location-based services in WLAN networks

In terms of Business Processes, LBS for Wi-Fi is mostly relevant in the area of marketing, service creation and delivery, and customer binding. It enables the creation of innovative application services. In the traditional 3G network architecture, value added services like user location are offered to 3rd parties by means of an OSA/Parlay gateway. Currently there are no standardised solutions to provide location information of users who roam between 3G and WLAN networks.

The sub-trial demonstrates a possible implementation for user location support in WLAN networks. With this implementation, the MNO can offer a single point of access for 3rd parties to obtain location information of roaming users. WLAN network operators can offer user location information to the MNO by means of a standardized Parlay X interface. The solution requires some management effort by the WLAN network operator. The location of the associated access point is used to pinpoint the location of the user. This means the network operator needs to administer the exact geolocation of the access points of the network.

4.2 Input requirements

The table below lists the requirements that have been used as a basis for the implementation of the sub-trial components. The requirements follow from [D7, ch.5] and [D8].

Requirement ID	Brief Description
SS2-1.0	<i>Support the roaming contract management process</i> The roaming contract management process composes of negotiation,

	implementation/ changing of roaming contracts. 3G OSS should offer concepts/tools to support the roaming contract negotiation and implementation/changing processes. The goal here is to automate these processes as far as possible without losing of the flexibility to re-design the processes whenever it's necessary.
SS2-1.1	<i>Support for inter-operations between MNO and WLAN-providers</i> There is the need of an extensible standard roaming inter-operational interface that enables all interactions between MNO and “little” WLAN provider while providing roaming services – in others words, a standard roaming inter-operational interface that implements the standard WLAN roaming contracts.
SS2-1.2	<i>Support for contract implementation/changing processes</i> A standard roaming inter-operational interface implements the standard WLAN roaming contracts. A standard roaming inter-operational interface implements the contract changing process. Implementation of changes, which often happen like tariff, should be kept efficient.
SS2-2.0	<i>Support for Authentication and Authorisation processes while user roaming</i> The focus here is on the roaming between WLAN – UMTS/GPRS. 3G OSS should offer functionalities so that home MNO can act as centralized operating role who is able <ul style="list-style-type: none"> ▪ to create authentication credentials for its roaming users, ▪ to distribute the credentials, ▪ to verify the credentials when a user tries to access a network, ▪ to revoke the credentials when needed (e.g. end of subscription, credentials stolen or otherwise compromised, etc.).
SS2-3.0	<i>Offer comprehensive tariff and cost information for users</i> Tariff and costs information should be showed to user on terminals and updated when user roams.
SS2-4.0	<i>Support LBS across different access networks</i> A 3 rd party application must be able to obtain a user's location from the network, without having to deal with network specific protocols or interfaces. To make it easier for 3 rd parties to offer these services in heterogeneous networks, the home network should unify the solutions of the different networks and provide a single point of access to obtain user location information. The following functions should be offered: <ul style="list-style-type: none"> ▪ Provide a service creation environment for 3rd parties, e.g. service discovery mechanisms, interfaces, and support. ▪ Provide authentication mechanisms for 3rd parties. ▪ Privacy: on user request, 3rd parties should not be allowed to retrieve the location information.
SS2-5.0	<i>Support discriminative seamless roaming concept</i>

	3G OSS should support interfaces offering information about current access networks and administration domains for implementation of discriminative seamless roaming. The decision to offer <i>which services in which network</i> will be made by the service provider.
SS2-6.0	Best connection 3G OSS should support the home MNO to collect and to analyse profiles of access networks, which users can use. With this information, the home MNO will be able to help users to select the “best” networks regarding cost and bandwidth. User should have possibility to define what means “best” for him.
SS2-7.0	Billing for Roaming 3G OSS should support the federated accounting concept that allows MNO and “little” WLAN providers to exchange billing information in a standardised and efficient way. The goal here is “one bill for all costs”.

The table below shows a mapping between the sub-trial requirements and the OSS components implementing these requirements:

Component	Restaurant Service	Q3ADE IPDR Store	Rating Bureau Service	Accounting Repository	Billing Engine	Interface Registry	AAA Server	AAA agent	AA Client	Domain Manager	UL agent	OSA/Parlay Gateway	Network Roaming Manager	Roaming Contract Manager	Mobility Client	Network Monitoring Tool
Requirement																
SS2-1.0														●		
SS2-1.1													●			
SS2-1.2													●	●		
SS2-2.0							●	●	●							
SS2-3.0																●
SS2-4.0	●					●				●	●	●	●		●	
SS2-5.0 (Note1)																
SS2-6.0																●
SS2-7.0		●	●	●	●											

Note 1: Discriminative roaming was not implemented. See section 4.7 for details.

4.3 Functional architecture

The requirements for this sub-trial have been implemented by creating a number of new OSS components, and re-using/enhancing some existing components from trial system 1. The functional architecture, as described in [D8], is evaluated from the perspective of the stakeholders that play a role in the scenarios as defined in section 4.3 of [D7]:

- Customer
- Mobile Network Operator (MNO)

- WLAN Network Operator
- 3rd Party Service Provider

4.3.1 Customer

From the customer (or end-user) perspective, the sub-trial fulfilled the requirements by demonstrating:

Non-interactive, secure WLAN authentication using digital certificates

As part of seamless roaming between heterogeneous networks, digital certificates were used to authenticate end-users to a visited WLAN, according to the IEEE 802.1x standard. This approach has a number of advantages, most notably the strong protection of user credentials, air interface encryption with regularly updated encryption keys, and the fact that authentication is automatically performed without the need for user interaction. A disadvantage for the customer is that the use of certificates is sometimes considered complex, although with the right tools and instructions (e.g. smartcard, USB dongle, dedicated client software) this should not be too problematic.

Obtain network characteristics (provider, technology, tariff, security, costs, data volume)

Customers get from a network monitoring tool all necessary information about her/his current access network environment like provider, access technology (LAN/WLAN/GPRS), current tariff, security level, cost as well as data load. This information is updated real-time in a configurable period of time. The implementation focused on proving the usefulness of this OSS functionality but not the scalability of the OSS component for commercial systems with a large number of users.

Automatic switching to the best network connection

Switching between network connections is based on the customer's preferences. Through a network service, delivered on the end terminal, customers can determine their preference (operator, cost, or access technology) for choosing the network connection they need. This network service chooses the network connection according to this preference when a customer roams through different access environments.

User location support in heterogeneous networks

For the roaming customer, the trial demonstrates the continuous availability of user location information, independent of the used network. In such a heterogeneous network environment, this enhances the user experience. The location-aware Restaurant Service is used as an example end-user service for this purpose. The concept is implemented and demonstrated using live WLAN, GPRS, and UMTS networks.

Privacy

In the trial, the Restaurant Service cannot access location information if the user does not allow it. The fact that the user can control these privacy settings is an important aspect of 3G OSS.

Billing

The concept of "one bill for network and service access", as demonstrated in the trial, is one of the main advantages of a 3G OSS for the customer. By collecting all charges for both network access and (3rd party) service access in a single bill, the customer does not have to deal with all parties separately for using these 3G services. The bill can be viewed on-line, and it is continuously updated with the latest charges.

4.3.2 Mobile Network Operator

In this sub-trial the Mobile Network Operator (MNO) plays the central role to provide 3G access networks and roaming services for its customers. The MNO establishes roaming contracts with WLAN providers, to allow its customers to use WLAN networks. Most of the implemented components in the sub-trial are components for the MNO's OSS.

From the MNO perspective, the sub-trial fulfilled the requirements by demonstrating:

Seamless roaming between fixed LAN, WLAN, and GPRS networks

This was already demonstrated in trial 1. The components implementing this function have been used as a basis for trial 2, and were extended to fulfil the trial 2 requirements.

Roaming Contract Management

The sub-trial offers MNO and WLAN providers the possibility to negotiate and create roaming contracts via a Web interface. The agreed roaming contract can be implemented in the OSS of MNO directly so that customer can start to use networks of WLAN providers. Tariff changing can also be carried out by a web interface and affects on the usage cost immediately after the negotiation session closes. The implementation focuses on the demonstration of the contract management process and keeps the contract's details simple intentionally.

WLAN authentication using digital certificates

The primary tasks for the MNO in the user authentication process are creation, distribution, verification, and revocation of user credentials. These credentials are used by roaming customers to get access to WLAN networks. When comparing the complexity of using digital certificates with – for example – username/password, there is no major difference for the MNO. With the right tools the handling of certificates can be automated. (see also section 4.6.2).

The biggest advantage for the MNO is that its roaming service offering can include automated, secure, seamless roaming for the end-user. Convenience for the end-user is always a good selling point.

Support for “best connection”

As mentioned in section 4.1, in spite of the fact that the interpretations of the term “best connection” can be quite different from the view of customers to the view of operators, the MNO still wants to offer this service as a value-added service. This network service has been implemented in the sub-trial by an OSS module, which is distributed both on the end terminal and on the core networks. The part on the terminal manages the customer access network preferences and performs the automatic network selection. Another part in the core networks collects the tariff and other network properties, needed for the network selection process.

Provide comprehensive tariff and cost information for end-users

With the software showing tariff and cost information on customer's end terminals, the MNO can carry out a service, as mentioned above, that is less in the interest of the MNO than in interest of telecommunication regulation and customer protection organisations. However these organisations are playing a role in the liberalised market.

User location support for heterogeneous networks

The trial demonstrates how user location interfaces from different network types can be unified into a single Parlay X user location interface, to be used by location-aware services. By implementing this concept, the MNO can strengthen its 3rd party service model. Service developers do not have to worry about the different networks a user can be in.

Billing

This sub-trial includes the same charging, mediation, rating, and billing components that are also part of the *3G Service Charging and QoS* sub-trial. However a different traffic accounting mechanism is used for charging of network usage. The Network Roaming Manager component monitors the amount of Mobile IP traffic for a user. The advantage of this approach is that no traffic accounting is needed in the access network. The MNO fully controls this process. A disadvantage is that an independent (WLAN) access network operator needs to trust the MNO to perform proper accounting for roaming users. This needs to be reflected in the roaming contracts. Another obvious disadvantage of this approach is that all traffic needs to be routed through the home network.

4.3.3 WLAN Network Operator

Seamless roaming

From the perspective of a WLAN operator, the seamless roaming concept represents an opportunity to attract a large user base without having to do marketing, maintain a subscriber database, or doing billing. By joining a seamless roaming architecture (through establishing SLAs with one or more service providers) a WLAN operator can focus on its core business: Running the WLAN network.

The seamless roaming architecture puts several requirements on the WLAN operator in order to properly support OSS functions such as authentication, charging, and billing. A lot of standardisation is still required in this area, mainly in terms of how existing standards will be deployed in practice.

User location support

To support value added services such as user location, additional software may be required at hotspot locations. In terms of requirements, and components to implement these, the WLAN operator will be most interested in the AAA agent and the UL agent AlbatrOSS components. For realistic deployment these probably need to be packaged as a single box (piece of hardware) and configuration and maintenance should be minimal. The trial shows the functionality that is required to support the various scenarios, and a way to implement this.

4.3.4 Third Party Service Provider

Value-added services

From a 3PSP point of view the roaming architecture offers access to new kinds of context information of users (e.g. location), which can be used to build new and attractive services. A key requirement for the 3PSP here is that this information can be obtained through standardized APIs which abstract from low level network details and are implemented in common technology. Parlay X (WSDL/SOAP) addresses this requirement. The trial demonstrates an example third party service (Restaurant service) that accesses such an API and obtains the user's location to show the nearest restaurants.

4.4 SID model

The SID Domains and information entities evaluated in this section are fully described in [D8]. Only the models relevant for this sub-trial are described.

The evaluation result for the Billing model is equal to sub-trial *3G Service Usage and QoS*. See section 3.4 for details.

SID Domain	Core Model: Service, Customer and Service Provider
Which entities are used?	User, ServiceProvider, Product, Service, Agreement, Location.
Which relationships are used?	Product-Service, Product-Location, Agreement-user, Agreement-product, Agreement-service provider.
Which attributes were added?	User: ID, (authentication) credentials. Product: ID, URL. Location: latitude, longitude, accuracy, measurement date/time.
Which entities/relations were not used?	Party, PartyRole, individual, organization, customer, SLASpec, SLAItem.
Justification for not using these entities.	For the purpose of the trial, these entities were either unnecessary or too abstract.
Which entities/relation were added?	-
Justification for adding these entities?	-
Should these entities be added to the SID model?	-

SID Domain	SLA
Which entities are used?	Service, agreement, Contract between MNO and WLAN-Provider. Roaming tariff.
Which relationships are used?	Agreement of MNO and WLAN provider.
Which attributes were added?	Roaming contract: ID MNO: ID WLAN-Provider: ID Tariff: ID of tariff
Which entities/relations were not used?	-
Justification for not using these entities.	-
Which entities/relations were added?	-
Justification for adding these entities.	
Should these entities be added to the SID model?	

SID Domain	User Preference
What domains are used?	User, Service
Which entities are used?	User access network preference
Which relationships are used?	User – MNO
Which attributes were added?	User: ID MNO: ID
Which entities/relations were not used?	-
Justification for not using these entities.	-
Which entities/relations were added?	-
Justification for adding these entities.	
Should these entities be added to the SID model?	-

4.5 OSS Components

4.5.1 Restaurant Service

A description of this component can be found in Section 4.4.3.2 [D9].

Relevant standards: Java, J2EE (JBoss), JAAS, WSDL/SOAP/XML (Axis), HTML/HTTP (Tomcat), and X.509.

The restaurant service shows the user's current location, and uses this information to locate relevant restaurants within the vicinity of the user. To obtain this location information, the RS uses the Parlay/X interface offered by the Domain Manager component (see section 4.5.10). There is flexibility in which server is used here per session: users from different providers typically have different domain managers. In addition, each DM hides the nature of the network the user is using to access the service.

The source of location information and the permission to obtain it depend on authentication of the end-user (and the RS itself), which illustrates a dependency between different OSS functions.

The nature of web pages (HTTP) influences the communication patterns and consequently interface definitions in the backend. The typical request/response model of the web implies a pull model for location information. To show an up-to-date map of the user's location, we added an automatic refresh to the web page, which triggers the browser to reload the page every X (15 in our case) seconds. For scalability reasons, it would likely be better to have an event-based location service in the network which sends updates to the restaurant service only when the user location changes by a certain minimum amount. This would reduce the amount of location requests (and subsequent map requests). However, pulling from the browser cannot be avoided unless proprietary mechanisms are used.

4.5.2 Q3ADE IPDR Store

This component is also used in the *3G Service Usage and QoS* sub-trial. See section 3.5.8 for additional information.

In this trial the Q3ADE IPDR Store is responsible for receiving usage records from the Network Roaming Manager and 3G Service (Restaurant) and mediate these into IPDR Records. The mediated IPDR Records are correlated based on service and session ID and forwarded to Rating Bureau Service as aggregated IPDR Documents.

4.5.3 Rating Bureau Service

This component is also used in the *3G Service Usage and QoS* sub-trial. See section 3.5.6 for the evaluation of this component.

4.5.4 Accounting Repository

This component is also used in the *3G Service Usage and QoS* sub-trial. See section 3.5.5 for the evaluation of this component.

4.5.5 Billing Engine

This component is also used in the *3G Service Usage and QoS* sub-trial. See section 3.5.3 for the evaluation of this component.

4.5.6 AAA Server

For a description of the AAA server, see section 4.4.3.5 of [D9].

The relevant standards and technologies used/implemented by this server are RADIUS, 802.1x, EAP/TLS, and X.509.

The three components AA client, AAA agent, and AAA server play a crucial role in the AlbatrOSS roaming scenario. Secure authentication is a key requirement when roaming between different network technologies, especially for wireless LANs. By automating the authentication process using digital certificates, the end-user experience is enhanced.

There is increasing industry support for wireless LAN authentication based on IEEE 802.1x, both from the Internet/IT world and telecom world.

The AAA server used in this trial was Lucent NavisRadius™, which is a Java-based, commercially available RADIUS server.

4.5.7 AAA agent

A description for this component can be found in [D9, section 4.4.3.4].

The relevant standards and technologies used/implemented by this component are RADIUS, EAP, and HTTP.

As part of the sub-trial WLAN access network, the AAA agent plays a dual role. It is primarily a RADIUS proxy server, transparently forwarding authentication request from

roaming users towards their home network AAA server. It does not interpret the EAP messages and has therefore no knowledge of the actual authentication mechanism that is used. It relies on the AAA server to do the actual authentication.

When access is granted by the AAA server, the AAA agent notifies the presence of the user in this access network to the UL agent, thereby triggering a location update event towards the Domain Manager of the user's home network.

Although it is a logical choice to couple the authentication of a user to the updating of location information, this coupling may be a disadvantage for other network technologies.

The sub-trial used FreeRadius as the basis for the AAA agent. This proved to be an extensible implementation, which suited the purpose of the trial, although it was a little cumbersome to actually implement the extensions. The main reason was the lack of documentation for this freeware product.

4.5.8 AA Client

A description for this component can be found in [D9, section 4.4.3.3]. The relevant standards and technologies used/implemented by this component are RADIUS, EAP, and HTTP.

The AA client fulfils client part of the 802.1x WLAN authentication solution. Modern operating systems like Microsoft Windows 2000/XP natively support it. The user interfaces and APIs that handle e.g. X.509 certificates are convenient and mature. When authentication and 3G/WLAN network selection functionalities are combined in one integrated client-side solution, the seamless roaming process can be fully automatic for the end-user.

A promising new development is SIM-based WLAN authentication. The authentication mechanism is similar to certificate based authentication, but the actual authentication is done by the 3G HLR.

Although this mechanism is still immature, standardisation is underway and initial experiments within AlbatrOSS have shown that it can be integrated easily into the seamless roaming concept.

4.5.9 Interface Registry

A description for this component can be found in [D9, section 4.4.3.1].

This UDDI registry is primarily used to decouple WLAN access network providers from the home network providers of roaming users. The home network providers register their location update interface in the registry. WLAN access network providers discover this interface when a roaming user enters the network.

Although UDDI is a standardized and useful solution for this type of service decoupling, the registration and discovery mechanisms are quite heavy for this purpose. In the sub-trial, the Interface Registry is merely used as a SOAP-based naming service. A more lightweight mechanism would be preferable, although SOAP as the basic information exchange protocol is considered useful from an interoperability point of view.

4.5.10 Domain Manager

A description for this component can be found in section 4.4.3.7 of [D9].

This component follows the Parlay X and OSA/Parlay user location specifications. Furthermore UDDI is used and HTTP servlets.

The Domain Manager has a dominant role within the user location concept demonstrated in the trial:

- Access networks (3G, WLAN,...) use it to notify the presence of a specific user within the network, and register the Parlay X URL where the user's location can be retrieved.
- Third party applications are able to obtain a user's location from the network, without having to deal with network specific protocols or interfaces.
- To ensure privacy, end-users are able to specify which applications can obtain their location and which cannot.

As the component interactions are mainly based on standardized Parlay X, the concept is suitable to deploy in a "real" network. The location update mechanism currently uses a proprietary servlet interface, this needs to be improved.

4.5.11 UL agent

A description for this component can be found in section 4.4.3.6 of [D9].

This component follows the Parlay X user location specification. Furthermore it uses UDDI and HTTP servlet technology.

The UL agent combines two main functionalities:

- a runtime part that detects and notifies the presence of a roaming user in the WLAN network,
- an OSS part to administer the IDs and locations of access points.

The advantages of having a user location solution in a WLAN environment are similar to the general advantages of OSA/Parlay: It offers the possibility to enrich the services, giving the network provider a marketing tool to attract more users and generate more traffic. The demonstrated concept is suitable for independent (enterprise or public) WLAN providers, as well as providers that are affiliated with for example mobile operators, e.g. WLAN hotspot providers.

The disadvantage of this specific solution is that there's a tight coupling with the authentication mechanism, which must be based on 802.1x. Furthermore, it is questionable whether the business model of having a lot of unaffiliated WLAN hotspot providers is realistic.

4.5.12 OSA/Parlay Gateway

A description for this component can be found in section 4.4.3.8 of [D9].

This component implements the OSA/Parlay and Parlay X user location specification.

The use of the OSA/Parlay gateway simulator in this sub-trial is restricted to the user location service. For the purpose of this trial it was extended with a Parlay X user location component.

The trial system includes one OSA/Parlay simulator, deployed in Lucent's test lab. During the trial execution, several AlbatrOSS partners used it simultaneously. At the same time also other projects and external parties used this simulator instance, without significant problems. The platform proved to be a stable OSA/Parlay simulation and testing tool.

4.5.13 Network Roaming Manager

A description of this component can be found in section 4.6.3.2 of [D9]. The relevant standard used by this component is the IETF Mobile IP protocol [RFC 2002].

As the “ground stone of the roaming platform”, the Network Roaming Manager implements 2 basic network services for roaming: handover (based on Mobile IP protocol) and best connection. Additionally, it collects data of resource usage and calculates the costs in real time. It contains a roaming centre database with diverse data of roaming partners, their relationships, their networks and networks capabilities.

4.5.14 Roaming Contract Manager

A description of this component can be found in section 4.6.3.2 of [D9]. The relevant standard used by this component is the IETF Mobile IP protocol.

This component has been implemented as a Web-service in the sub-trial. MNO and WLAN provider can perform contract management activities like negotiation/creation, change, delete. The roaming contracts are contained in the roaming centre database.

4.5.15 Mobility Client

A description of this component can be found in section 4.6.3.5 of [D9]. The relevant standard used by this component is the IETF Mobile IP protocol.

The Mobile Client is the “remote agent” of the Network Roaming Manager on customers’ terminals. It works together with the Network Roaming Manager to perform:

- The handover process between different access networks, and
- The best connection service.

Additionally, the Mobility Client manages the customer’s profile containing the customer access network preferences. The Mobility Client is combination of development of T-Systems Nova and a Secgo Mobile IP client.

4.5.16 Network Monitoring Tool

A description of this component can be found in [D9, section 4.6.3.3].

This component brings the transparency regarding the cost, and access environment for customer. It receives information from the Network Roaming Manager concerning the actual location of the mobile terminal, the access technology, tariff, cost and show them on the terminal’s display.

The Network Monitoring Tool is a Java application with further outlook at the adaptation for different types of end terminals.

4.6 Implementation Technologies and Standards

4.6.1 Technologies

The technologies used in this sub-trial are largely identical to the technologies used in the *3G Service Charging and QoS* sub-trial. See section 3.6 for evaluation details.

4.6.2 Standards and Recommendations

Parlay X

The Parlay X user location API [ParlayX] is used to provide a web services based mechanism to retrieve location information from heterogeneous networks. Although the standard – like SOAP/WSDL on which it is based – is immature, it is proven to be convenient and useful for this purpose.

A drawback of the Parlay X specification is that no access control mechanism is defined for application services, in contrast to “traditional” OSA/Parlay. This is out of the specification’s scope, and consequently the implementation needs to decide what the best access control mechanism is for the situation. Depending on the deployment environment (public/private, Internet/Intranet based, ...), a heavy or more lightweight mechanism may be required.

The sub-trial did not implement a Parlay X access control mechanism. This is for future research.

Mobile IP

The standard protocol accepted for supporting macro-mobility within the IP networks is Mobile IP with its two versions: MIPv4 [RFC 2002] and MIPv6 [MIPv6]. These protocols deal with device mobility, i.e. the migration of terminal or routers from one link to another but do not consider the possible migration of code over physical nodes, or so called *computational mobility*. The protocols are standardized by the IETF.

RADIUS

The IETF RADIUS protocol [RFC 2865] is used as part of secure WLAN authentication. In this trial, it is mainly used as an encapsulation protocol for EAP messages, as part of the IEEE 802.1x access control mechanism [802.1x]. The standard is mature, extensible, and useful for this purpose.

On the downside, RADIUS has some security flaws. For example, it is known to be vulnerable to denial of service attacks and session hijacking, and it does not implement a replay protection mechanism. Higher layers like EAP can correct some of these flaws, but operators may still be reluctant to deploy RADIUS in a public network environment.

[DIAMETER], the successor of RADIUS, addresses most of these issues. This protocol is however not yet fully standardized, nor is it commonly used yet.

EAP/TLS

The EAP protocol is used as part of the 802.1x access control mechanism. It is used to encapsulate TLS protocol messages, which in turn is used to exchange X.509 digital certificates between AA client and AAA server. Both EAP and TLS are off-the-shelf technologies that are suitable for this purpose.

X.509

This ITU recommendation defines the structure of the digital certificates, which are used to authenticate users when they enter a WLAN access network. This is a powerful mechanism, which makes a fully automated authentication possible, thereby making the roaming experience *seamless*. Furthermore the authentication is secure, with a strong protection of the user credentials.

A drawback of using digital certificates is that it requires a public key infrastructure (PKI). The creation, distribution, and revocation (e.g. in case of theft) of certificates is not a trivial task and requires expertise. These processes can however be automated. Once this is in place, it can be a very powerful authentication mechanism.

4.7 Deviations

Discriminative Seamless Roaming has been described as a concept in [D7], but not demonstrated the sub-trial. The focus of the sub-trial is on the network services and the demonstration services “Restaurant Service” and “MeetMe” are not suitable to demonstrate this concept (not “discriminative”-able enough). However, the ground stones of this concept have been implemented. MNO can offer SP information about the owners and the characteristics of every access networks.

The concept of “Roaming contract on the fly” has not been realised in the sub-trial because we recognised that there is less opportunity for such a business case.

The creation and distribution of X.509 client certificates has been described as a business process in [D7] and as functionality in [D8], but there was no component part of the component architecture implementing it. However an on-line service was created as part of the test environment to create a client certificate on the fly. This forms the basis of the process.

Revocation of digital certificates is not part of the demonstration scenario.

The sub-trial functional architecture as laid out in [D8] includes the WLAN Metering component, which is not included in the execution phase of the trial. It was recognised that the function did not add any value to the concepts of this sub-trial, which uses Mobile IP based traffic metering. Furthermore identical functionality was already part of the *3G Service Charging and QoS* sub-trial.

Finally, as already noted in section 3.7, the Mobile MeetMe service was replaced by the Restaurant Service.

4.8 Conclusions

The *Inter-technology and Inter-domain Roaming* sub-trial demonstrates a number of concepts that are important in a liberalised 3G market. Mobile network operators, WLAN access providers, 3PSPs, and possibly other players need to cooperate to create the best possible services for roaming users.

The following main concepts are prototyped and demonstrated in this sub-trial:

- Seamless roaming between heterogeneous networks, with an emphasis on secure authentication
- Roaming Contract Management
- Provide network tariff information to end-users
- Best connection network selection
- User location support in heterogeneous networks
- Single bill for network and service usage

The implementation and integration of the components that perform these functions has been done using standard technologies and tools. The IP protocol and the Internet form the basis for all management related communication between OSS components.

IETF Mobile IP is used as a basis for most of the sub-trial scenarios. The fact that a roaming Mobile IP user always registers at home when entering a visited network, makes a centralized

approach possible for functions like traffic metering, pushing network properties, location updating, etc. A disadvantage of Mobile IP is that part – or all – of the network traffic is routed through the home network (although some operators may consider this an advantage).

The sub-trial uses web-based technologies like SOAP and HTTP servlets to exchange management data between most of the components. This is considered convenient in a prototyping environment, but may be less suitable in today's 2G and 3G mobile networks. However, the convergence between mobile/fixed telecommunication networks and the Internet makes these technologies considerably more significant.

5 Evaluation results: Delivery of Integrated Services

This sub-trial intends to demonstrate the service aggregation concept, which states how service providers build their end-user services by putting together several 3rd party services. It shows how the service provider, by aggregating all these services, is able to deliver integrated and end-to-end managed services to end-users. The DIS sub-trial relies on a process-oriented NGOSS architecture with a process manager as a backbone of the OSS platform, interacting with the different components over a well-defined Web service interface.

The component architecture and interfaces of the DIS sub-trial are explained in [D9], section 5.3. Further information about the functional architecture of the sub-trial can be found in [D8] and also in [D6] section 2.

5.1 Business processes support

Service Design

The Service Design aims to integrate existing components based on static views and dynamic views enabling to deploy and operate the service very rapidly and to automate the corresponding processes. The Service Design step does not include the development of components. The Service Design step allows mapping of business requirements for new services into models which will be essential elements for implementing and operating the service.

This step is based on the building of sequence diagrams, specifying how the different service components integrate with each other to conform a new one and its interaction with the Process Manager to achieve the flow diagram necessary to the implementation of this new service. And to reflect what the requirements are and how they should be addressed in order to be fulfilled by the different components of the system.

Service Implementation

Service Implementation step is processed after the Service Design step and aims to get the service ready to be deployed. Therefore after Service Design step, some models need to be mapped to specific formats before being instantiated into the different components.

Service implementation is achieved in the DIS sub-trial by the Service Creation environment, which enables the implementation of the business logic and the modelling of the service and service components. The process is carried out in a very easy manner from the GUI tool provided which allows to re-design or change some parts of the service models and very easily re-implement the service or some parts of it.

Service Instantiation

Service Instantiation is the first step of deployment process in order to make the service ready for users subscriptions and then to be activated for usage.

The service instantiation is first achieved by the deployment of the different WS components in the application server. And then, by the injection of the business process logic for the new services into the Process Manager, as the injection of the programming code necessary.

Subscription

The service subscription introduces the first interaction with users (subscribers in this process) and needs a user session with a login phase (we assume that there is a previous process where the customer at the first time gets a login and password to access to the service provider platform and where his user profile is previously created).

The subscription process is achieved through the User Subscription Manager component and the User Profile Manager. Users can subscribe to a new service by selecting the service name in a subscription GUI and the service quality level he needs. Finally, the user has to confirm all selected items and for this a summarised contract GUI is proposed.

Service Activation

Service Activation step is processed just after the Service Subscription step and aims to get the service ready to be used.

The service activation process in our sub-trial consists of the customer notification confirmation. Also, for quality management purpose, some events like activation start and activation end are tracked and sent to activation performance collection component.

Usage

The service usage supports a complete user session from login, to logout. The role of the service usage process flow is to describe how the different involved components interact while the service is being provided to the user.

In our sub-trial, we distinguish the phases of “Identification and Authentication”, where a simple login/password mechanism has been implemented, “Usage”, operated by the process manager, which must interact with other business flows and “Logout”, which implements the process that terminates the service usage and trigger specific operations.

Data Collection, Analysis & Control

The data collection, analysis & control process is a very sensitive OSS process. Many other business processes (assurance, billing) are dependent on it. This process is in charge of collecting the different data that will be used:

- To charge the user based on service use (usage data).
- To evaluate the service performance and quality and achieved the required repairing actions (performance data).

The DIS sub-trial separates collection from analysis & control. The latter will be achieved through other process flows (service quality management, reporting, etc.). Therefore, this section focuses on the collecting aspect.

The data collection process flow interacts with the following components/process flows:

- The Resource Repository Manager to store the collected data,
- Service Usage, which provides the user id for which performance (for data that are user dependent) and usage must be measured.

Service Quality Management

The Service Quality Manager process monitors service quality in real time. It is based on SLA definition from activation phase to SLA life-cycle end. During this time, the Service

Quality Manager process monitors, notifies billing and maintenance components for every SLA degradation or violation and provide service quality reports.

This business process is deployed in our sub-trial through the Service Quality Manager (Hewlett Packard component), which collects performance indicators from other components and produces Service Quality Indicators. These data are compared with corresponding degradation and violation thresholds defined in Service Levels. It manages to provide the user with quality reports.

Reporting

The reporting process flow defines the different actions that must be achieved after a given period (periodic reporting) or at the end of a service session (session reporting) to report service-related information (charging, billing and performance) to customer or user.

The SID sub-trial focuses on the reporting at the end of the session (called the session reporting), and several components are involved in the process, such as the Billing, Service Quality Manager and Resource Repository Manager components. The session reporting process flow generates a session report.

5.2 Input requirements

The tables below list the requirements that have been used as a basis for the implementation of the sub-trial components. The requirements follow from [D7, ch.5] and [D8].

Service lifecycle step	Req. ID	Service Aggregator Requirements
SERVICE DESIGN	DIS-1.0	<i>Get methodology guidelines for supporting Service Designers</i> Part of the Service Creation Environment (SCE) to be provided to Service Designers includes a set of guidelines allowing them to adopt common practices for designing services. Moreover, service design should be based on object-oriented approach and therefore should allow reusability of existing Service Component (in the same way a design pattern for OO development).
	DIS-2.0	<i>Get tools supporting and automating service designers work</i> Based on requirement for rapid service creation and update, the SCE should provide tools as much as possible. Main requirements for these tools are: (i) to use existing modelling languages (UML, WSDL, BPEL4WS, ...), (ii) to be integrated at the user level, i.e. providing one main GUI allowing reuse of the model between the different tools, (iii) to be integrated with other lifecycle steps (service deployment).
	DIS-3.0	<i>Get the possibility to reuse service design patterns from one service to another</i> It is quite important to capitalise from the service creation process and one way is to get the possibility to reuse already implemented service components, in the form of design patterns and the possibility to integrate them into the design of new services.

	DIS-4.0	<p><i>Ease the mapping/integration of business requirements into service models</i></p> <p>The Service Design should start the design of services from the definition of Business Processes and Business Models. These ones are based on the business vision of the service and therefore enclosed the business requirements. On the other side it is necessary to map these business entities from existing and constraining existing components (networks, systems, applications). This mapping will be facilitated if the <i>existing components</i> are defined in the form and based on the same ontology as the business ones.</p>
	DIS-5.0	<p><i>Support integration into the overall supply chain (3PSPs)</i></p> <p>The SCE needs to integrate the fact that Business Models include more and more actors and relationships between these actors for the provisioning of services. Therefore it is necessary to get the possibility to integrate <i>external service components</i>.</p>
	DIS-6.0	<p><i>Enable automation of the process flow</i></p> <p>The interactions between the different components are defined as the process flow. One aim of the service design is to capture the different possible flows between the components and to define this at business level in term of business logic. This should ease integration of service components and it should provide the possibility to play with the different components for different scenarios at business level.</p>
	DIS-7.0	<p><i>Enable rapid changing and creation of services</i></p> <p>Service Providers and particularly Service Aggregators will have to create a lot of services in a very short time. They will also need to be able to adapt existing services to the demands and to specific users (personalization). This will be a main criterion for evaluating SCE and accompanying tools.</p>
	DIS-8.0	<p><i>Support aggregation of services into an integrated service provided by the service provider and 3PSPs</i></p> <p>Aggregated Services are based on the possibility to integrate service components, defined in the same way as COTS, into one entity. Service Aggregators' main particularity will be to provide access to a large range of services to its customers. These services will be based on 3PSP content and/or applications.</p>
Service Implementation	DIS-9.0	<p><i>Get tools to support and automate service designers' work</i></p> <p>The Service Implementation step is processed after the Service Design step and aims to get the service ready for deployment. Therefore, after Service Design, some models need to be mapped to specific formats before instantiation into the different components. The Service Implementation should be supported by tools enabling smooth integration between Service Design and Service Implementation. The Service Implementation step should be fully automated based on Service Design results. Service Implementation tools should be configured based on the targeted service execution and operation environment.</p>

	DIS-10.0	<i>Automate the mapping of formats and coding of models</i> The mapping of formats and coding of models should be done automatically and launched from Graphical User Interface.
	DIS-11.0	<i>GUI for the control of a step</i> Even if this step is mainly an automated one, errors can be generated during service design. We need the possibility to easily update the service design to remove any errors.
	DIS-12.0	<i>Possibility to re-implement the service or some part of it</i> Service Providers should have the possibility to redesign or change some part of the service models and very easily re-implement the service or some part of it.
Service Instantiation	DIS-13.0	<i>Service Deployment phase should be automated as much as possible</i> Service Instantiation corresponds to the main step of the Service Deployment phase. Deployment of a service is today a time-consuming activity. This is mainly due to the fact that we cannot adopt a modelling approach yet. Such modelling could result, at the deployment stage, in the mapping and integration of models to the specific OSS Business components.
	DIS-14.0	<i>Possibility to update existing services</i> Services, for a lot of different reasons, need to be updated, and therefore we need to be able to re-deploy a service that has been updated at Service Design stage.
Subscription	DIS-15.0	<i>Possibility to authenticate the users with maximal security</i> For online subscription the Service Provider needs maximal security regarding the customer (e.g. non repudiation).
	DIS-16.0	<i>Possibility to subscribe to new services should be presented automatically to users, based on user preferences</i> The user should be aware when new services are offered to customers, and should be able to subscribe to these services in a very convenient way. Services proposed to the users should take the user preferences into account.
	DIS-17.0	<i>Service Provider should be able to automate the SLA negotiation process</i> When customers subscribe online to services they will be able to select a service level. This selection should be validated on the SP side, based on available service resources and other constraints. In case the SP cannot accept the service level, a negotiation will take place. If such a process could be automated, time could be gained.
	DIS-18.0	<i>After SLA has been concluded all necessary information should be passed to the Service Activation step automatically</i> A new SLA should be stored into an SLA repository and an indication should be sent allowing activation of the service based on user requirement.
Service Activation	DIS-19.0	<i>Automate the service activation tasks</i> The service activation tasks correspond to the enforcement of the SLA, which means that the service will be configured based on customer requirement. These tasks need to be automated to be able to offer personalized services to customers.

Service Usage	DIS-20.0	<p><i>The user gets access to content and applications from different 3PSPs in a transparent way</i></p> <p>The Service Aggregator provides the possibility to access a large range of content and applications provided by different 3PSPs. From the user perspective these services are accessed from one focal point without knowing about the location of 3PSPs. Applications and content also need to be adapted to the user context.</p>
	DIS-21.0	<p><i>The user get access to services in a secured way</i></p> <p>The user is authenticated when accessing services. Information exchanged between SAP (Service Access Point) and the user can be encoded when necessary.</p>
	DIS-22.0	<p><i>The service level is assured to users based an SLA defined with him</i></p> <p>The Service Aggregator has to insure the service level as defined in the SLA during service usage.</p>
Data Collection Analysis & Control	DIS-23.0	<p><i>The Data collection must be reliable (no failure is accepted) as it is a sensitive process</i></p> <p>The data collection, analysis & control process is a very sensitive OSS process. Many other business processes (assurance, billing) depend on it, and failure could have bad consequences on the overall processes.</p>
	DIS-24.0	<p><i>The collected data must be accurate, particularly the usage data</i></p> <p>It is necessary to insure accuracy of information collected and to be able to demonstrate this accuracy.</p>
Service Quality Management	DIS-25.0	<p><i>Service Quality Management should allow to monitor overall and customer specific QoS (against SLOs as defined in SLAs) in real time</i></p> <p>Service Quality Management should provide information in real time and operators should be alerted when service quality is being degraded based on defined objectives. Service Operators should also be able to evaluate the impact on the customers.</p>
	DIS-26.0	<p><i>QoS degradation should be reported with identification of the customers impacted and elements for isolating the problem rapidly</i></p> <p>Service Quality Management should indicate the impact of QoS degradation to customers. It should provide also the possibility to identify the origin of the problem.</p>
	DIS-27.0	<p><i>A SLA violation should not be reported to a user who did not suffer from it</i></p> <p>SLA violation should be reported to customers only when users depending on this customer have been impacted.</p>
	DIS-28.0	<p><i>Service Quality Management should provide the possibility to map resource performance information (KPI) into service quality indicators (KQI)</i></p> <p>The Service Quality Management should provide mechanisms that allow mapping of resource performance indicators into service related quality indicators.</p>

Reporting	DIS-29.0	<i>The reporting process should provide 2 modes for reporting information to the operators and customers: on-demand, scheduled reporting</i>
	DIS-30.0	<i>Service Quality report should contain all elements allowing to validate SLA has been respected</i> Service Quality reports should give the user the possibility to insure the SLA has been respected
	DIS-31.0	<i>Invoices should contain all service usage elements, when requested by the user, as well as details on the tariff applied</i> It is quite important for a customer to understand what he has to pay and how the bill has been calculated.

The following table presents the Customer requirements:

Service lifecycle step	Req. ID	Customer Requirements
SUBSCRIPTION	DIS-31.0	<i>The user should be identified and authenticated by the service</i>
	DIS-32.0	<i>The user should have access to his/her subscribed services</i>
	DIS-33.0	<i>The user should have access to all services and services levels for subscription</i> The user should be able to subscribe to new services and to select the service level expected when the service is used.
Service Activation	DIS-34.0	<i>The user should be notified for service activation</i> The user should be informed when a service has been activated and he can get access to it.
Service Usage	DIS-35.0	<i>The user should be identified and authenticated by the service</i>
	DIS-36.0	<i>The user should have access to his/her subscribed services.</i> The user should get access to subscribed services based on conditions defined in the SLA.
	DIS-37.0	<i>The user should get an address of his/her current position</i>
	DIS-38.0	<i>The user should get the map related to his current position</i>
	DIS-39.0	<i>The user should get flight information when requested</i>
	DIS-40.0	<i>The user should get web-sites that concern his/her current position and a particular field of interest</i>
	DIS-41.0	<i>The user should be able to control the way he/she can be reached through his/her mobile) during the travel</i>
Service Quality Management	DIS-42.0	<i>The user should be notified in case of service quality degradation or SLA violation</i>
Reporting	DIS-43.0	<i>After a session the customer should be informed of the potential SLA violation during the last session and the cost of this session</i>

The table below shows a mapping between the sub-trial requirements and the OSS components implementing these requirements:

Requirement	User Interface Manager	User Security Manager	Session Manager	User Profile Manager	Data collection Manager	Customer Reporting Manager	Service Quality Manager	User Subscription Manager	Process Manager	OSA Parlay X	Mapporama	Google	AirFareQuote	Across Communication	InstantMessagingService
DIS-1.0															
DIS-2.0									•						
DIS-3.0									•						
DIS-4.0									•						
DIS-5.0									•						
DIS-6.0									•						
DIS-7.0									•						
DIS-8.0									•						
DIS-9.0															
DIS-10.0															
DIS-11.0															
DIS-12.0									•						
DIS-13.0									•						
DIS-14.0									•						
DIS-15.0		•													
DIS-16.0								•							
DIS-17.0								•							
DIS-18.0								•							
DIS-19.0							•		•						
DIS-20.0	•														
DIS-21.0		•													
DIS-22.0							•								
DIS-23.0					•										
DIS-24.0					•										
DIS-25.0					•										
DIS-26.0					•										
DIS-27.0					•										
DIS-28.0					•										
DIS-29.0						•(*)									
DIS-30.0						•									
DIS-31.0		•				•									
DIS-32.0	•			•											
DIS-33.0	•							•							
DIS-34.0							•								
DIS-35.0		•													
DIS-36.0	•							•							
DIS-37.0	•									•	•				
DIS-38.0											•				
DIS-39.0													•		
DIS-40.0												•			
DIS-41.0														•	•
DIS-42.0						•	•								
DIS-43.0						•									

(*) partly: periodic report has not been implemented

5.3 Subtrial architecture

In general, the architecture fulfilled the functional requirements as stated in [D7], plus other NGOSS and non-functional requirements.

The architecture is evaluated from the perspective of the following stakeholders:

- End-user
- Service provider
- 3rd Party Service Provider

5.3.1 End-user

This subsection describes the architecture evaluation results from the perspective of a end-user. The end-users subscribe and access the to the Travel Assistant service, which provides them with Location Based services, Flight schedule information and Location Based Web Search.

The evaluation results of OSS functions that are relevant to the end-user are as follows:

Reporting

The reporting mode that has been implemented for the DIS sub-trial is a post-session mode that enables to provide end-users with immediate information regarding the cost the last session. The solution implemented for the DIS sub-trial relies on the UHC mediation component (section 3.5.8), the WIT billing engine component (section 3.5.3) and the ATOS Customer Reporting Manager component (section 5.5.6). When the session ends, the CRM calls the billing engine to generate charging information regarding the last session. The billing engine calls the IPDR store components to retrieve data collected during the last session. The trial demonstrates the interest and the feasibility to obtain real-time charging information and to provide the end user with them, through a NGOSS process. The technical limitation was the following:

- The polling time requested by the IPDR Store requires interrupting the reporting process for 30 seconds. The IPDR Store has to detect the end of session event in the collected data to generate the record with duration measurement that will be retrieved by the billing engine. This point shows the kind of problem that can occur when synchronising different web services to support a general process.
- It was not possible to provide the end-user with performance data, as this function is not supported by the SQM.

User Security and Privacy Management

The User Authentication and Identification is achieved during the login process by the USPM component. The trial demonstrates the interest to separate the user authentication process into a specific component that can focus on several security processes to make sure that the user is granted permission for accessing the different services. It proved also the feasibility of supporting authentication operation. However this issue was not a priority of this sub-trial. Next evolutions of the USPM would implement:

- Encryption algorithms.
- Certificate-based authentication.
- Policy-based security management.

Session Management

A session is supported by the Session Manager and the Process Manager components. As soon as the Process Manager detects a login attempt, the Session Manager creates a session. The Session Manager provides an ID to the session, manages the state of the session (not implemented) and manage session problems if any (exception during the session). Other issues like the session life-cycle management and synchronisation of components are addressed by the generic usage process. The session manager was not a priority component of the DIS sub-trial, however the session management is an important OSS function that provides reliability to the whole system and the key role of processes to support session management in a flexible way has been proved by the trial.

5.3.2 Integrated Service provider

Integrated Service Provider (ISP) is in charge of issues such as user's location, availability and provision of location-based services, security and adaptation.

The evaluation results of OSS functions that are relevant to the Integrated Service Provider are as follows:

User Profile Management

The User Profile Management business process enables storing and retrieval of user related information like:

- Civil status.
- Contact information (postal address, email, phone).
- Accounting information.
- Personal information.
- Service preferences.
- PAM information.
- Agreement information (Service level, contract, subscribed services...).
- Etc...

The User profile description relies on the AlbatrOSS SID model that has been implemented into a MySQL database. In the DIS sub-trial, the PAM profile management has been defined as part of the user profile. It enables the user to set his preferences for the different agents he possesses, as recommended by the OSA/Parlay working group. Business processes enable to get the user profile (to retrieve information about the user), and to exploit the PAM information in a specific scenario (notification of urgent information).

Data Collection Analysis and Control

The data collection analysis and control process is one of the key business processes that have been selected to demonstrate the advantages of the service-based architecture for OSS. Two modes have been implemented to support this process:

- The first mode: merge the data collection process with specific service usage processes (which is an inconvenient): Each data collection operation is explicitly called by the process that orchestrates the related service.
- A Process implicit mode: This mode relies on the handler mechanism provided by axis web services that enable to activate a specific process when a service operation is called (before calling, after calling). This process used a specific XML file, that provides the types of measurement that must be applied (volume-based, duration-based, event-based) and parameter(s) that must be stored. This mode is

more complex to implement than the previous one but enables a weak coupling between the data collection process and the service process itself.

The trial showed the interest of both approaches: they provide a flexible mechanism to support data collection that can be modified very easily, either by changing the process or the XML file. This point is very important for integrated service providers who must be able to modify quickly their offer according to business opportunities. The collection process relies on the IPDR store component used by other sub-trials, which plays very well its mediation role by taking into account the specific DIS sub-trial data format.

Service Quality Management

This process is another important OSS process selected by the DIS sub-trial. When an end user service depends of availability and performance of 3rd party services, it is important to have contractual mechanisms that include Service Level Agreement. The service aggregator is responsible of providing added value services to end-users and he must have guarantees from his suppliers.

It has been decided to rely on the HP SQM product that provides an innovative solution to manage service quality. The trial shows the interest of service-oriented architecture to integrate separate solutions on an OSS platform. In the current implementation of the DIS sub-trial the SQM component notifies the HP SQM product using TIBCO messages when a new service is subscribed by a given user and must be activated. SQM plays the role of a gateway between the albatross system and the SQM component, translating data, concepts and terms from the AlbatrOSS world to the HP SQM world.

The HP service level models have been applied to the specific context of integrated services. This model can be applied to any kind of integrated service that puts together specific business components (billing, etc...), connectivity services and content services. This model enables to manage an end-to-end service quality by offering an intuitive process to aggregate atomic QoS parameters. The trial showed the relevancy of this generic model, and has been proposed to HP to support aggregated services.

User Subscription Management and activation management

The activation process does not require a dedicated component. It is completely driven by a NGOSS process that notifies the different involved components. On the contrary, the subscription process relies on a specific component called the subscription manager that concentrates all the subscription related functions. This observation shows the designer dilemma when designing a service-oriented architecture: finding out the right balance between processes and components, to provide the most flexible and reusable solution. Service oriented architectures are particularly relevant to support activation processes that require many interactions with different components addressing separate issues (billing, SLA, user information, etc...).

Process Management

The process management makes concrete the concept of service aggregation. Separate 3rd party services, which provide rough content and/or connectivity, can be put together to obtain a final, end-user, and added-value service. The trial demonstrates the value of this approach, by selecting four existing web services: Maporama, Google, AirFareQuote and Across Communication and building first interesting functions of the travel assistant end-user service. The trial proved also the value of the architecture (service oriented architecture) and the technologies (web service and process manager).

5.3.3 Third Party Service Provider

3PSPs are aggregated by the Integrated Service Provider, which builds different added-value services by putting together several 3rd party services.

The evaluation results of OSS functions that are relevant to the 3PSP are as follows:

Billing

The Service Aggregation concept proposes a new service value chain with different stakeholders in new business environments with new business models. The billing solution must be able to support these new environments and models. Particularly, the billing solution must be able to support fast changes and dynamic adaptation of the value chain. For instance, if the integrated service provider adds new services or changes some suppliers, renegotiates tariffs with suppliers, the billing solution must be flexible enough to take into account the new billing schemes. The trial points out the interest of such a solution.

Process Management: see 5.3.2.

Service Quality Management: see 5.3.2.

5.4 SID model

Core domain:

SID Domain	Core: user, service providers, products
How SID has been implemented?	Implementation into a mySQL database with some simplifications to keep the demo simple.
Which entities are used?	User, ServiceProvider, Product, Service, Agreement
Which relationships are used?	Product-Service, Agreement-user Agreement-product Agreement-service provider
What attributes were added?	Product: ID, name, URL. User: id, name, lastName, msisdn, loginName
Which entities/relations were not used?	Party, PartyRole, individual, organization, customer, SLASpec, SLAItem, Location
Justification for not using these entities.	We don't use SLA related entities because SQM manages a different model. We don't use Party and other related entities to have a simple model for the demo. Location has been defined as an attribute of the Product entity (URL)
Which entities/relation were added?	None.

PAM:

SID Domain	PAM
Which domains are used?	Implementation into a mySQL database A version as EJB entity beans is available also (first trial).
Which entities are used?	All: the PAM domain of the AlbatrOSS SID model is already a simplification of the ones provided by the

	OSA Parlay PAM subgroup.
Which relationships are used?	All
Which attributes were added?	None.

Billing

The billing part of this sub-trial is equal to the *3G Services Charging & QoS* sub-trial. See section 3.4 for SID model details.

SLA

The DIS sub-trial relies on the HP SQM product to provide quality of Service management facilities. This product possesses an internal data model to represent QoS related entities. For this reason the DIS sub-trial did not use the SID model SLA domain. However an approximate mapping can be done between both models, as mentioned in the following table:

SID Model SLA domain	HP SQM model
Service	Service
Product	Service Group
Customer	Customer
Service Level specification	Service level
Agreement	SLA (Service Level Agreement)
Service Level Spec Objectives	SLO (Service Level Objective)
Service Level Spec Consequence	Action
Service Level Specification parameters, KeyQualityIndicatorSLSParam	Performance Parameters

The table above indicates that most of the SLA entities defined into SID Data Model are defined also in the HP SQM model which confirms the relevancy of the AlbatrOSS SID model, as HP SQM is a product that relies on the same conceptual hypothesis.

5.5 OSS Components

5.5.1 Component User Interface Manager (UIM)

This component is described in section 4.1.3.10 [D9].

It translates the user actions into process manager events and provides functions that can push information to the user. It provides an interface between the servlet environment, and the Process Manager.

The UIM has been implemented as an AXIS (Apache) v1.1 web service.

5.5.2 User Security and Privacy Manager (USPM)

This component is described in section 4.1.3.8 [D9].

Further description of the component can be found in section 4.3.2.3.1[D8].

For security reasons, the SAP Manager does not share user authentication information with other components. Therefore the SID does not cover this aspect.

The USPM has been implemented as an AXIS (Apache) v1.1 web service.

5.5.3 Session Manager (SM)

This component is described in section 4.1.3.8 [D9].

Some Processes run by the Process Manager overlap some functions of this component, which makes it less useful.

The SM has been implemented as an AXIS (Apache) v1.1 web service.

5.5.4 User Profile Manager (UPM)

This component is described in section 4.1.3.6 [D9].

Further description of the component can be found in sections 6.3.2.2 and 4.3.2.3.2[D8].

Two versions of the UPM exist:

- One directly connected to a MySQL database.
- Another connected to the Resource Repository Manager.

The UPM has been implemented as an AXIS (Apache) v1.1 web service and it relies on a mySQL database that implements the AlbatrOSS SID model.

5.5.5 Data Collection Analysis and Control Manager (DCACM)

This component is described in section 4.1.3.9 of [D9].

Further description of the component can be found in sections 6.3.2.7 and 4.3.2.3.7 of [D8].

This component collects performance data and usage data by using two modes:

- A process explicit mode: Each data collection action is explicitly defined into the process that orchestrates the related service. The inconvenience of this approach is that the service process is not separated from the data collection process.
- A process implicit mode: The way data are collected for a given service is described in an XML file that provides the type of measurement that must be applied for each service operation (volume-based, duration-based, event-based) and data that must be collected. This collection mode enables to separate specific service orchestration issues from data collection issues.

The DCACM has been implemented as an AXIS (Apache) v1.1 web service. Its functions rely on the eTOM recommendations.

5.5.6 Customer Reporting Manager (CRM)

This component is described in section 4.1.3.7 [D9].

Further description of the component can be found in sections 6.3.2.6 and 4.3.2.3.6[D8].

This component will be in charge of periodic reporting (bill generation, performance report) in a later version.

The CRM has been implemented as an AXIS (Apache) v1.1 web service.

5.5.7 Service Quality Manager (SQM):

This component is described in section 4.1.3.2 [D9].

Further description of the component can be found in sections 6.3.2.5 and 4.3.2.3.5[D8].

The SQM component provides an interface between the DIS AlbatrOSS platform and the Hewlett Packard's product HP SQM.

The SQM has been implemented as an AXIS (Apache) v1.1 web service. It uses the TIBCO message-based solution (version 4.1) to send messages to the HP SQM product during subscription and activation processes.

5.5.8 User Subscription Manager (USM)

This component is described in section 4.1.3.12 of [D9].

Further description of the component can be found in sections 6.3.2.3 and 4.3.2.3.3 of [D8].

The USM has been implemented as an AXIS (Apache) v1.1 web service and interacts with the mySQL database containing the AlbatrOSS SID Model.

5.5.9 Process Manager (PM)

This component is described in sections 4.1.3.4 and 4.5.3.5 of [D9].

It relies on the TID workflow engine, based on the use of Petri Networks. The Process Manager has been implemented as an AXIS (Apache) v1.1 web service.

5.5.10 Billing Engine

This component is also described in AlbatrOSS D8 section 6.1.2.10 and D9 section 4.8.3.3. In this sub-trial the Billing Engine is responsible for providing the Customer Reporting Manager with rated service usage information in the form of IPDR records so that detailed billing information about customer activity can be prepared.

5.5.11 Q3ADE IPDR Store

In this sub-trial the IPDR Store is responsible for retrieving usage records from the Data Collection Manager (DCM) in a DCM specific format, and transforming these into IPDR records. During mediation the DCM specific records are transformed into the general PAR structure used in the other sub-trials.

The Q3ADE IPDR-Store performs correlation of usage records within each service, based on service session ID. The correlated IPDR records are forwarded as IPDR Documents to the Rating Bureau Service.

Further description of the component can be found in section 6.1.2.11 [D8] and section 4.7 [D9].

5.5.12 OSA/Parlay X

The OSA/Parlay gateway, with a Parlay X extension, generates location information of end-users that are part of the trial scenario. The location information is used by the integrated 3rd party service (travel assistant). The same component is used as in the *Inter-technology and Inter-domain Roaming* sub-trial. See section 4.5.12 for details.

5.5.13 Third-party services (non OSS components)

Maporama Web Service: This component is described in section 4.1.3.16 and 4.1.3.17 of [D9].

Google Web Service: This component is described in section 4.1.3.14 of [D9].

AirFareQuoteService: This component is described in section 4.1.3.15 of [D9].

InstantMessagingService: This component is described in section 4.1.3.18 of [D9].

Across Communication Web Service: This component enables to send SMS to a given user. All information is provided at the following URL: <http://www.acrosscommunications.com>

5.6 Technologies and standards

5.6.1 Technologies

5.6.1.1 Servlet Technology

The user front-end part of the UIM is implemented with the servlet technology. Servlet technologies are considered to be more efficient, convenient, powerful, portable, inexpensive than many alternative CGI-like technologies:

- **Efficiency:** with CGI, each new HTTP request needs to start a new process, and the time for launching this process can be quite long. With Servlet, the Java Virtual Machine remains active, and each request is managed by a lightweight Java thread. Moreover, from a memory management perspective, traditional CGI needs to load N times the same CGI program if there are N requests that must be handled by this CGI program. Servlet Technology requires N threads, but only one Servlet class. Servlets provide useful optimisation mechanisms like memory caching and keeping database connections open.
- **Convenience:** Servlet is convenient, particularly for Java programmers, who do not need to learn other script languages like PERL, etc...Servlet offers dedicated and “programmer friendly” API to:
 - Parse and decode HTML,
 - Read and set HTTP headers,
 - Handle cookies,
 - Track sessions,.....
- **Powerfulness:** Servlet technology provides Java programmer with embedded mechanisms that enable to do several things that are more difficult with traditional CGI:
 - Accessing to the web server environment and required resources stored in standards place (like images),
 - Sharing data between different servlets,
 - Easier Connection to database,
 - Maintaining data from requests to requests (session tracking).
- **Portability:** As a Java program Servlet can be executed on different Servlet containers.
- **Inexpensive:** Some free solutions exist to support the Servlet environment on the server side (Apache/Tomcat). Moreover for commercial web server solutions, the addition of servlet facilities is generally free of charge.
- **Flexibility:** Servlets are independent entities that handle particular requests from users. This relative low coupling level with other servlet instances provides the integrated service provider with a flexible way to build the front-end environment.

Regarding the DIS sub-trial, and the role of the integrated service provider, even if the servlet side does not represent the core technology implemented by this sub-trial, it is interesting to notice that the servlet technology provides integrated service providers with a flexible way to build its front end. Figure 1 shows a kind of design pattern that can be applied to easily manage the addition/removal/change of new third party services in the back end.

The GUI part of a given service is defined as a “*service GUI set*” composed of

- a set of servlets that manages user requests and responses to them,
- an abstract representation of the service functionalities that defines all the functionalities supported by the service from a user point of view,
- an abstract web page factory that will generate the most appropriate format (HTML, WAP ,etc...) regarding the related device,
- a set of concrete web page factories for each device type that requires a specific adaptation.

This “*service GUI set*” is plugged to the User Interface Manager through one implementation of the abstract service representation. Each time a new service is added, the related *service GUI set* is implemented on its side and plugged to the UIM. The removal of a service consists only in removing the service GUI set from the platform.

The Servlet **atomicity** makes this kind of deployment a lot easier.

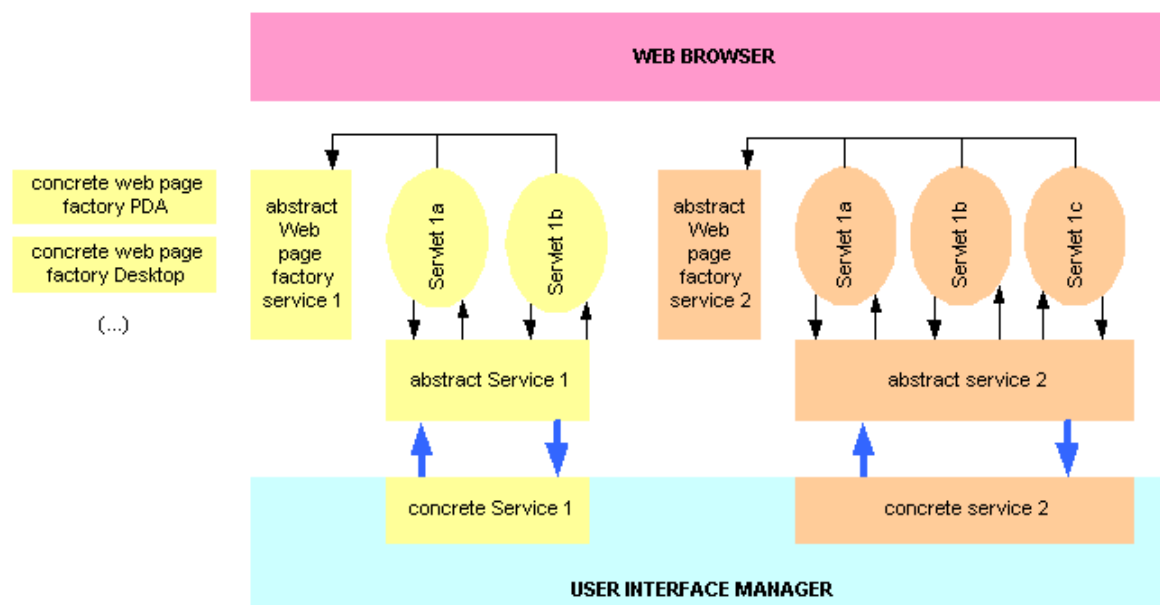


Figure 1 The service GUI set

One of the most annoying problems with servlets is the necessity to generate the HTML syntax on fly, directly in the Java code, which is not really convenient, particularly when debugging the system. The possibility to define HTML templates, like in JSP, would be better. Moreover servlets require compilation, as they are not based on a script language. The servlet deployment requires defining a particular XML file (web.xml) where all the servlets must be declared with mapping URL, etc. These constraints make the servlet environment more difficult to deploy than other technologies.

5.6.1.2 Apache Tomcat

Apache Tomcat is a free servlet server provided by the Apache Jakarta Project, which provides open source solutions relying on the Java technology. Tomcat is the servlet container that is used in the official reference implementation for the Java Servlet and JavaServer pages technologies. Apache Tomcat is easy to install and servlets can be easily deployed through the .war file that contains the different Servlets and utility classes, as well as the web.xml file that provides servlet configuration information. Though, Tomcat requires to be restarted each time

a modification is made to the different Java classes (servlets, other utility classes), which is rather annoying when debugging the system.

However, regarding the technology evaluation aspect of the DIS subtrial, it does not really make sense to evaluate Tomcat as a servlet server. Tomcat is only used to interact with the user and from this point of view, it does not have a key role in the architecture defined to support integrated services.

5.6.1.3 Web services

The different points that we want to evaluate are:

- Is it possible to implement an OSS infrastructure using the web service framework?
- Can we meet NGOSS recommendations with web service implementation concepts?
- Can an OSS component be implemented over a web service?

To summarise, web services can be defined as the sum of the following technologies:

- A protocol that enables to support data exchange between components: SOAP.
- A format to describe component interfaces: WSDL.
- A mechanism to support “white page” and “yellow page” service: UDDI. We did not use this mechanism in our sub-trial.
- An orchestration mechanism that enables to define and run business processes over a set of components: We used the proprietary solution of TID: NGOSS platform.

The evaluation of web services will be achieved against these 3 aspects: SOAP, WSDL and TID NGOSS platform.

5.6.1.4 SOAP

The Apache/Axis web service server, which has been used to implement most of the OSS components, made the W3C SOAP protocol transparent. SOAP requests and responses are managed through stubs that can be automatically generated by some Axis-dedicated commands. However, SOAP brings good interoperability facilities, as all the Axis components are able to communicate together and with other web services implemented on other platforms. The performance issue can be raised, particularly the time required to serialise/deserialise java objects into SOAP descriptions. This time is critical in an integrated service environment, where several services can be cascaded to provide the final result to the users. For instance, we were required to increase the timeout threshold at different places of our architecture to avoid irrelevant timeouts. The evaluation of this time should be extended to the general communication time evaluation over Internet, in order to make sure that this time is compatible with OSS time constraints.

5.6.1.5 WSDL

For each component a WSDL interface has been provided. This WSDL interface provides the different operations and related messages required to communicate with the component. This WSDL description can be generated from the different components contractual definitions described into the D8. However, WSDL syntax does not allow representation of pre-conditions and the post conditions, which is a limitation of the contract-programming paradigm specified by the TM FORUM NGOSS.

5.6.1.6 TID NGOSS platform

The orchestration of the different web services with different business processes is achieved by using the TID NGOSS platform. This platform provides tools to design and execute the different processes that enable different OSS components and service components to interoperate. A Petri network is used to model the process where each state enables to call a particular web service operation, and to wait for a list of event occurrences that determine the next state. The Process Manager works in an asynchronous mode, which has the following consequence: in a given state the PM cannot wait for a Web service operation result. Thus, some wrappers are required to enable to the PM to interact with Web service components. Moreover the PM is implemented over a CORBA solution.

Pros:

The main advantage of TID's NGOSS solution comes from its ability to support the NGOSS recommendations like:

- Apply distributed system design principles.
- Apply COTS system technologies.
- Separation of the business processes from the functional components.
- Data sharing and stewardship.
- Component programming by contract.
- Run-time discovery/integration of services.

As far as the integrated service provider role is concerned, these ways of programming OSS provides many advantages like:

- Better capability to integrate easily separated 3rd party services into the same added-value service.
- Better capability to add/remove a 3rd party service to improve the added-value behaviour.
- Better capability to change the 3rd party service suppliers when required.
- Separation of the business processes (fulfilment, billing, assurance) from the service specific processes and factorisation of the business processes.
- Better implementation and testing processes. Each component is implemented on its side. Then tested with unitary processes. Then the integration takes place with integration tests.
- Better integration of all the components.

Cons:

We can mention the following problems regarding the NGOSS platform used:

Synchronous / Asynchronous mode:

Web services use a synchronous interaction mode as well as the TID Process manager requires an asynchronous mode. This technical constraint has two consequences on our OSS platform:

- Web service wrappers were required to make them able to communicate with the PM. The implementation of these wrappers is quite easy and an automatic generation process can be provided later on. However the necessity of these wrappers makes the web service integration a little bit more complex and increases the communication time between components.
- The UIM that manages the interaction between the user and the Process manager must also be able to translate synchronous interactions into asynchronous ones. Again this constraint weights the GUI implementation down, by requiring particular synchronism

mechanisms and parameter passing. For instance any function addition to the GUI requires implementing a function to send a related event as well as another function to get back the results.

Processes as Network of networks:

To keep an evolutionary solution, the different processes have been designed as separate Petri sub-networks that are connected through a Meta Network (network of networks); this process organisation has some consequences on the way the events are sent by the UIM to the Process manager. This raises some design problems, as it is difficult to keep a loose coupling between the GUI and the rest of the system. A possibility to limit this issue exists, for instance by providing a PM wrapper between the UIM and the PM, but the schedule was too tight to implement it.

Synchronisation between the GUI and the Process Manager:

The state of the GUI must be perfectly synchronised to the state of the PM. Again this constraint increases the coupling between the GUI side and the PM side and makes the GUI implementation much more complex than in a classical client-server solution. For instance, any user action must have a related state defined into the Petri network process, and must generate the precise expected event.

CORBA serialisation /deserialisation:

The process that has been selected by TID to serialise/deserialise Java objects over TID's CORBA infrastructure is rather painful and not sustainable. Other mechanisms that enable to pass Java objects into an XML strings will be provided in a next version.

Comparison to other orchestration solutions:

A comparison of the TID solution to other orchestration solutions would be useful: Candidate solutions are WSDI (decentralised approach) and BPEL4WS (centralized approach) that are still not mature but can bring interesting alternatives in the future.

5.6.2 Standards and recommendations:

5.6.2.1 UML

The different processes have been specified and designed using sequence diagrams. UML sequence diagrams enable to show the different interactions between the components, differentiate the synchronous from the asynchronous mode, define the different input and output parameters, and provide the different events that must be sent by the UIM to the Process Manager. The collaborative design of the different sequence diagrams for the different involved processes makes the integration of the different components and processes much easier. The different objects exchanged by the different components at process runtime, were modelled with class diagrams. UML appears to be the essential complement of any distributed approach.

5.6.2.2 eTOM

Many processes and/or components that have been implemented to support the DIS OSS platform were defined into the eTOM TM FORUM framework. It enables to capture in the same picture all the Telecom OSS infrastructure related processes. eTOM provided very relevant guidelines to the DIS subtrial definition particularly when defining the integrated

service life cycle and specifying the related business processes. The DIS subtrial selected from eTOM the following business processes:

- **Service creation (eTOM Service development and retirement):** The end-user service is created by aggregating and synchronising different service components (OSS components: Billing, QoS, etc... and 3rd party service components).
- **Service deployment (eTOM Service development and retirement):** The service is made available on the 3rd party service provider platform for subscription.
- **Service subscription (eTOM fulfilment – Customer Interface Management):** The customer subscribes the related services.
- **Service activation (eTOM fulfilment -Service Configuration and Activation):** After subscription, the service is provided to the user.
- **Authentication:** The user is authenticated and identified by the system.
- **Data collection (eTOM data collection Analysis and Control for billing and assurance):** Usage and performance data are collected to support billing and assurance.
- **Rating (eTOM billing specific instance rating):** The service usage is rated according to a tariff table and the customer is charged according to its rating.
- **Service quality management (eTOM Service Quality Management):** Through Service Quality objectives, the Different service level agreements are permanently monitored and any degradation or violation is *detected*.
- **Reporting (Assurance: Service quality analysis action and reporting and billing):** Service Usage (post session reporting) is reported as well as service performance (SQM monitoring).

eTOM focuses on the business aspect of telecom issues and does not address service related issues like service delivery (how the service can be delivered to the user taking into account his device, the user's context, the roaming aspects...).

5.6.2.3 NGOSS

Main requirement addressed by the DIS sub-trial concerned integration of the different platforms. Each platform used different technologies and communication paradigms. Integration of the different components in such an environment has proven the necessity for the definition of an infrastructure supporting easy plug in of the component. Therefore, it has been necessary in the second phase for this sub-trial to build an OSS infrastructure capable of supporting an easy integration of new components, based on NGOSS/TMForum architecture principles:

NGOSS Core Principles:

Apply Distributed System Design Principles

Components have a total independence from each other since they work as isolated Web Services, keeping their business logic internally. The integration of these components comes through the use of the Process Manager, which holds that individual business logic and contains the control logic, which orchestrates the business logic.

Apply COTS (Commercial Off-The-Shelf) System Technologies

Components can be obtained in any way: Developed, loaned or purchased, and then integrated in an independent way in the infrastructure and with no added costs.

Separate Process Flow from Application Component Operation

It is achieved through the use of a Process Manager, which integrates each component business logic through a control logic (network of networks).

Provide Shared Information Services

The Process Manager owns a shared information view of data interchanged within the components, so it is responsible to provide the data in a distributed way (service).

Apply Common Communication Services

Achieved thanks to the use of a SOAP logic bus. Components communicate through this protocol since they are implemented as Web Services.

Define OSS Functionality with Contract Interfaces

By means of the use of WSDL as the contract specification language.

Run-time Discovery/Integration of Services

The discovery of new services was thought to be implemented through the use of UDDI as the way to publish new services. The integration of new services was considered through the separation of the control logic from the business logic in the Process Manager. This allows us to add new services without affecting the existing ones, both in their development and deployment.

NGOSS Extended Principles:

Common form of Systems Management

By means of the Web Services deployment tool, plus the new services deployment tool of the Process Manager. Also, management of network elements is performed in an external way.

Common form of Systems Security

Achieved thanks to the use of a security and/or privacy management component, which allows/denies the access to the system functionality.

Business/System Definition Linkage

This principle was not addressed.

Linkage to other systems that are not NGOSS compliant

Achieved through the service integration via the Process Manager, any system can be linked as a new component wrapped or accessed through an adapter.

Linkage into Legacy / Heritage Systems

It applies the same as for the previous NGOSS principle.

Support for Customer Service Management (e.g. customer self-serve, B2C)

By means of the Process Manager web tool for the components management, which allows the services dynamic deployment.

Model Driven Approaches (e.g. SLA/QoS definitions for Service Fulfilment and Assurance)

Achieved through the use of business models based on Petri Networks, which allow the modelling of the flow of each of the services to be deployed, as well as the Process

Manager business logic. Afterwards, these services can be mapped into specific models depending on the technology.

5.7 Deviations

Functional deviation:

All the business processes that have been defined into the deliverable D8 and D9 have been implemented. Some processes have been simplified:

- Reporting process: No periodic reports are generated.
- Data collection process: Two processes are available:
 - An explicit process that calls directly the data collection manager during the service usage,
 - An implicit process that uses an XML file to decide which data must be collected according to the called service operation. This process has been simplified.
- PAM management process: The process that enables to update the PAM profile has not been implemented at the moment.

All the business components that have been defined into the deliverable D8 and D9 have been implemented. Some of these components have been kept simplified:

Reporting Manager: It does not deliver periodic reports.

Session manager: It does not support all the reliability functionalities (failure, session interruption, etc...).

Service creation manager: Only a simple prototype has been implemented.

Notification Service manager: Only manages mail and SMS (not voice).

Design deviation:

We implemented a complete solution based on the separation of the business processes (loaded on the TID platform) and the business components (as web services).

Some deviations concern mainly the basic service components:

- Resource repository manager: It does not support one-to-many relations and complex queries.
- No naming service has been implemented and / or integrated (UDDI).
- No event service has been implemented.

About the service quality management, we only managed a limited set of QoS parameters:

- Network Latency.
- ResponseTime of 3rd party services.
- Operational State at the integrated service level.
- Service availability percentage at the integrated service level.
- Operational state, ServiceAvailability of 3rd party services won't be measured as planned in the D8/D9.

5.8 Conclusions

The DIS sub-trial demonstrated the advantages provided by an infrastructure which facilitates the quick and easy deployment of value added services by the Service Providers. The sub-trial also demonstrated the Service Creation environment, which enables the implementation of business logic and modelling the service and service components.

The scenario executed in the DIS sub-trial was defined from two perspectives, the end user's and the Integrated Service Provider. The Travel Assistant service was presented, where the user subscribes and accesses to this location-based service, which results from the integration of several basic services provided by 3PSPs.

6 Trial 2 Summary and Conclusions

The primary objectives of the second phase of trials were to enhance the systems developed for the first trial and to develop new components to fulfil the missing functionalities identified in the evaluation of the first trial. The results of the evaluation of the first trial provided inputs to the AlbatrOSS Intermediate Architecture and Specifications [D8] and the second phase of trials were based on the subsequent realisation of this architecture. It should be noted that the enhancements were not purely technological or functional, but were highly dependant on the requirements emanating from the emerging business models identified.

This document has presented an in-depth evaluation per trial-sub-system. The comprehensive evaluations have been conducted under a myriad of headings ranging from business process support to technologies and standards used. This approach was taken to target as wide an audience as possible, from business analysts to technical developers that operate in the OSS space and beyond.

Three sub-trial systems were evaluated. The OSS topics chosen by sub-trial systems were:

- 3G Services aggregation and dynamic management
- Service product provisioning via a portal
- QoS management
- Usage-based charging and convergent online billing
- Flexible and federated mediation of network and service usage
- Inter-technology and inter-domain roaming

These topics demonstrated a set of well-defined OSS functions while following three customer scenarios. The customer scenarios involved provisioning of 3G services and highlighted the requirements on the OSS.

The main aim was to evaluate functional architecture, SID information model, and OSS components developed in the second iteration of AlbatrOSS project. The general idea was to take the stakeholders involved in the AlbatrOSS business model and storyboard and evaluate the architecture from their perspective. The evaluation is qualitative and the main criteria are (1) OSS functions (2) requirements (3) information entities derived from SID (4) development technologies used (5) industry and international standards used.

The general conclusions that can be drawn from the trial evaluations is that the trials have addressed many of the primary issues arising from the increased levels of cooperation required between stakeholders in the emerging business models. A brief examination of each of the trials highlights this fact. Distributed Integrated Services, Convergent Billing, Service Portals, Inter-Domain Roaming, Inter-Technology Roaming, End to End QoS etc. all relate back to the overlying business principle of inter provider cooperation. The AlbatrOSS "One Stop Shopping" paradigm has shaped the Architecture [D8] which in turn has influenced the aforementioned attributes of the various trial sub-systems. Hence, the project partners conclude that the trials have met the vast majority of their objectives in relation to the AlbatrOSS 3G OSS architecture and business requirements derived from the project activities.

The results of this evaluation process will be used to influence the final version of the AlbatrOSS Architecture that will be captured in Deliverable D10 in the coming months.

7 References

- [D7] AlbatrOSS, *Business Model and Scenarios definition (Version 2)*, IST-2001-34780 Project Deliverable 7, May 2003.
- [D8] AlbatrOSS, *Intermediate Architecture and Specifications*, IST-2001-34780 Project Deliverable 8, July 2003
- [D9] AlbatrOSS, *System Version 2 ready for Trial 2*, IST-2001-34780 Project Deliverable 9, October 2003
- [GB921] enhanced Telecom Operations Map (eTOM): *The Business Process Framework For The Information and Communications Services Industry*, TM Forum Approved/Version3.0 June 2002.
- [GB 923] *Wireless Service Measurements Handbook*, GB 923, Version V2.0, TM Forum, April 2003.
- [ITU-T Y.1541] ITU-T Y.1541, SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE AND INTERNET PROTOCOL ASPECTS, *Internet protocol aspects – Quality of service and network performance, Network Performance Objectives for IP-based services*, May 2002.
- [ITU-T I.380] ITU-T I.380, SERIES I: INTEGRATED SERVICES DIGITAL NETWORK, *Overall network aspects and functions – General network requirements and functions, Internet protocol data communication and functions – IP packet transferred and availability performance parameters*, February 1999.
- [RFC 792] *Internet Control Message Protocol*, September 1981.
- [RFC 2002] *IP Mobility Support*, Standards Track, October 1996.
- [MIPv6] *Mobility Support in IPv6*, Internet-draft, June 2003.
- [RFC 2330] *Framework for IP Performance Metrics*, Informational, May 1998.
- [RFC 2680] *One-way Packet Loss Metric For IPPM*, Standards Track, September 1999.
- [RFC 2681] *Round-trip Delay Metric For IPPM*, Standards Track, September 1999.
- [RFC 2865] *Remote Authentication Dial In User Service (RADIUS)*, Standard Track, June 2000.
- [RFC 3148] *A Framework for Defining Empirical Bulk Transfer Capacity Metrics*, Informational, July 2001.
- [802.1x] IEEE, *Port-based network access control*, IEEE Std 802.1X-2001.
- [DIAMETER] *Diameter Base Protocol*, Internet-draft, December 2002.
- [SID] TeleManagement FORUM, *Shared Information/Data (SID) Model: Concepts, Principles, and Domains*, Member Evaluation Version 2.0, December 2002.
- [SID-4SO] TeleManagement FORUM, *Shared Information/Data (SID) Model: Addendum 4SO – Service Overview Business Entity Definitions*, Member Evaluation Version 1.0, November 2002.
- [SID-1A] TeleManagement FORUM, *Shared Information/Data (SID) Model: Addendum 1A -Common Business Entity Definitions – Agreement (including Service Level Agreement)*, Member Evaluation Version 1.5, December 2002.
- [3GPP 21.905] 3GPP, *Technical Specification Group Services and System Aspects: Vocabulary for 3GPP Specifications*, 3GPP TR 21.905 V6.0.0 (2002-09) (Release 6).
- [3GPP 23.107] 3GPP, *Technical Specification Group Services and System Aspects: Quality of Service (QoS) concept and architecture*, 3GPP TS 23.107 V5.6.0 (2002-09) (Release 5).

[3GPP 23.207] 3GPP, *Technical Specification Group Services and System Aspects: End-to-End Quality of Service (QoS) concept and architecture*, 3GPP TS 23.207 V5.5.0 (2002-09) (Release 5).

[ParlayX] Parlay Consortium, *Parlay X API specification*, available from <http://www.parlay.org/specifications>.

[IPDR311] *Network Data Management Usage for IP-Based Services*, version 3.1.1, October 9, 2002.

[J2EE] *Java 2 Platform, Enterprise Edition (J2EE) Specification*, version 1.4, November 24, 2003.

[XML DB] *XML DB API Working Draft*, September 20, 2001

[XML RPC] *XML RPC Specification*, June 15, 1999

Annex 1 Acronyms

3GPP	3 RD GENERATION PARTNERSHIP PROJECT
3PSP	THIRD PARTY SERVICE PROVIDER
AA	AUTHENTICATION AND AUTHORISATION
AA	AUTHENTICATION, AUTHORISATION AND ACCOUNTING
AlbatrOSS	ARCHITECTURE FOR LOCATION BASED APPLICATIONS OF THIRD GENERATION OPERATION SUPPORT SYSTEMS
BPEL4WS	BUSINESS PROCESS EXECUTION LANGUAGE FOR WEB SERVICES
CAN	CONTENT/APPLICATION NETWORK
CORBA	COMMON OBJECT REQUEST BROKER ARCHITECTURE
COTS	COMMERCIAL OFF-THE-SHELF
CSC	CUSTOMER SERVICE CONTRACT
EAP	EXTENSIBLE AUTHENTICATION PROTOCOL
EDN	EXTERNAL DATA NETWORK
FMA	FEDERATED MEDIATION ADAPTER
GUI	GRAPHICAL USER INTERFACE
HLR	HOME LOCATION REGISTER
HTTP	HYPERTEXT TRANSFER PROTOCOL
ICMP	INTERNET CONTROL MESSAGE PROTOCOL
IETF	INTERNET ENGINEERING TASK FORCE
IPDR	INTERNET PROTOCOL DETAIL RECORD
IPPM	INTERNET PROTOCOL PERFORMANCE METRICS
J2EE	JAVA 2 PLATFORM ENTERPRISE EDITION
JAAS	JAVA AUTHENTICATION AND AUTHORIZATION SERVICE
JDBC	JAVA DATABASE CONNECTIVITY
KPI	KEY PERFORMANCE INDICATOR
KQI	KEY QUALITY INDICATOR
LHP	LMIF-IN-HTML PROTOCOL
LMIF	LOCAL MANAGEMENT INTERFACE LANGUAGE
MAC	MONITORING ANALYSIS COMPONENT
MAN	MOBILE ACCESS NETWORK
MNO	MOBILE NETWORK OPERATOR
NDM-U	NETWORK DATA MANAGEMENT-UTILIZATION
NGOSS	NEW GENERATION OPERATIONS SYSTEMS AND SOFTWARE

NPE	NETWORK PERFORMANCE EVALUATOR
ODBC	OPEN DATABASE CONNECTIVITY
OSS	OPERATION SUPPORT SYSTEM
PAR	PROPRIETARY ACCOUNTING RECORD
PING	PACKET INTERNET GROPER
PKI	PUBLIC KEY INFRASTRUCTURE
PM	PROCESS MANAGER
RADIUS	REMOTE AUTHENTICATION DIAL-IN USER SERVICE
RBS	RATING BUREAU SERVICE
RPC	REMOTE PROCEDURE CALL
RTD	ROUND TRIP DELAY
RTT	ROUND TRIP TIME
SDU	SERVICE DATA UNIT
SID	SHARED INFORMATION AND DATA
SIM	SUBSCRIBER IDENTITY MODULE
SLA	SERVICE LEVEL AGREEMENT
SLO	SERVICE LEVEL OBJECTIVES
SOAP	SIMPLE OBJECT ACCESS PROTOCOL
TLS	TRANSPORT LAYER SECURITY
UIM	USER INTERFACE MANAGER
UML	UNIFIED MODELING LANGUAGE
USIM	UNIVERSAL SUBSCRIBER IDENTITY MODULE
W3C	WORLD WIDE WEB CONSORTIUM
WLAN	WIRELESS LOCAL AREA NETWORK
WSDI	WEB SERVICES DYNAMIC INVOCATION
WSDL	WEB SERVICE DESCRIPTION LANGUAGE
XDR	XML DATA REDUCED
XML	EXTENSIBLE MARKUP LANGUAGE
XML:DB	XML DATABASE
XML-RPC	XML REMOTE PROCEDURE CALL
XSLT	EXTENSIBLE STYLESHEET LANGUAGE TRANSFORMATIONS