OMA Service Provider Environment Requirements

Candidate Version 1.0 – 19 May 2009

Open Mobile Alliance

OMA-RD-OSPE-V1_0-20090519-C

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

Use of this document is subject to all of the terms and conditions of the Use Agreement located at http://www.openmobilealliance.org/UseAgreement.html.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavours to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the "OMA IPR Declarations" list at http://www.openmobilealliance.org/ipr.html. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE "OMA IPR DECLARATIONS" LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

Figures

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

Tables

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

1. Scope (Informative)

This specification defines the use cases and requirements for the OMA Service Provider Environment (OSPE). The OSPE addresses the need for a standardized environment/infrastructure for developing and deploying services, and addresses the need to quickly and efficiently identify and diagnose faults associated with services that are offered by a Service Provider.

Lower service life cycle costs can be achieved by reducing administration (service and subscription provisioning, O&M, etc.) and integration costs of the components used to construct services. In addition, service manageability can be improved by defining service level diagnostics to enhance the Service Provider's ability to determine whether their complex services are working correctly and if a fault arises in any part of the service chain, that this fault is easily and efficiently traceable and rectifiable.

The key objective for the OSPE is to fulfill the following design principles:

- Allow component interchangeability;
- Allow the multi-vendor mixing-and-matching of components by identifying their interfaces and having consistent semantics of shared data/schema across these components. To achieve such replaceability/reusability of components, we must provide mechanisms to support the full life cycle of components. This can be achieved by identifying standard component interfaces to handle aspects such as installation, configuration, administration, versioning, etc.

Note that since OSPE is a Reference Release, the architecture will identify the interfaces, but there will be no interface specifications as part of the OSPE Reference Release.

OSPE will focus standardization efforts in the following areas:

- Service Level Diagnostic capabilities including identifying interfaces between components and other Service Provider environments;
- Capabilities and identifying interfaces to achieve life-cycle management, including plug & play, of components, applications and services.

Section 5 of this document contains use cases describing the issues related with enabler development, deployment, integration, administration, maintenance, etc. Section 6 contains requirements extracted from the use cases presented in section 5.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

2. References

2.1 Normative References

[OMADICT] "Dictionary for OMA specifications", Open Mobile Alliance™, OMA-Dictionary-V2_1-20040914-A, URL:http://www.openmobilealliance.org/

[OSE] "OMA Service Environment", Open Mobile Alliance™, OMA-Service_Environment-V1_0-20040907-A, URL:http://www.openmobilealliance.org/

[Privacy] "Privacy for Mobile Services requirements", Open Mobile Alliance™, OMA-RD-Privacy-V1_0-20031104-C, URL:http://www.openmobilealliance.org/

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[TMF] "TM Forum Glossary",TeleManagement Forum™

URL:http://www.tmforum.org/

2.2 Informative References

[ARCHREQ] "OMA Architecture Requirements", Open Mobile Alliance™, OMA-RD_Architecture_V1_0-20031021-A, URL:http://www.openmobilealliance.org/

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Definitions contained in [OMADICT] are not repeated.

Application	See description in [OMADICT].
Catalogue	A logical collection of information associated to services, components, and applications and interfaces deployed in the OSPE.
Component	A replaceable/reusable unit in the service provider system that is responsible for a particular set of functionality and associated information. A component forms part or all of an enabler. See also the description in [OSE].
Control Plane	The control plane provides for the initiation, termination, management, maintenance and supervision of connections.
Device	A device is a voice and/or data terminal used for information transfer. Device types may include (but are not limited to): mobile phones (GSM, CDMA, 3GSM, 802.11 etc.), data-only terminals, PDAs, laptop computers.
Enabler	See description in [OMADICT].
End user visible event	An events that is visible to the end-user consuming a service. A user-visible event may be a diverted-to-number, Browsing-Redirect URL, MMS Content-to-person, Messaging Notification of delivery or submission.
Execution environment	The set of application programming interfaces (APIs) and semantic constraints in which program logic executes.
Marked device/component	A device or component that has been marked and has not been unmarked.
Marking	The operation of enabling a device or component to subsequently initiate the Service Level Tracing.
Marking request	A request initiated by the Service Provider or other authorized actor, which marks a device or component. The marking request contains the SLT start trigger event.
OMA Service Provider environment	A system that consists of applications and multiple standardized components that are used to construct end-user services. Examples of an OSPE may include an Enterprise network, Mobile network, or Third Party Service Provider.
Plug and Play	A capability of an entity to recognize the introduction of a new entity and make use of services offered by that new entity, or to provide services to the new entity, without human intervention [TMF].
Publish	The process involved with making information available about a service, component, application, or interface available to its intended audience, when they have been deployed in the Service Provider environment.
Quality of Service	The collective effect of service performances that determine the degree of satisfaction of a user of the service [TMF].
Registration	The process of providing information related to a service, an application, a component, and interfaces in the Service Provider environment.
Resource	A resource is any component, enabler, function or application that can receive and process requests. A Resource is an abstract concept that represents a capability, e.g. a network element, in a Service Provider’s domain [OSE].

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

	Resources represent physical and non-physical components used to construct Services. They are drawn from the Application, Computing and Network domains, and include, for example, Network Elements, software, IT systems, and technology components [TMF].
Service	See description in [OMADICT].
Service chain	A concatenation of components that are used to support a service. A Service chain may cross multiple Service Provider domains.
Service deployment	The process of deploying a service in the service provider's environment. It includes all steps needed to get the service up and running and ready to be subscribed by end-users (from the service installation to the final tests and service activation).
Service Enabler	See description in [OMADICT].
Service level agreement	A formal negotiated agreement between two parties, sometimes called a Service Level Guarantee. It is a contract (or part of one) that exists between the Service Provider and the Customer, designed to create a common understanding about services, priorities and responsibilities [TMF]. In OMA the term Customer is used to represent the Service consumer.
Service level tracing	Service Level Tracing is the ability to capture and log all relevant information at each component within a service chain, associated with a specific service that is initiated either by an end user or a component.
Service life cycle	The process a service goes through from idea, to creation, to introduction in the Service Provider environment, to retirement (when a service is removed from the Service Provider environment).
Service Package	A Service Package consists of services grouped under one commercial package or bundle and that is offered to customers (e.g. end-users).
Service provisioning	The configuration of service software (i.e. setting of its system-wide parameters). Service provisioning is part of the Service Deployment.
Service subscription provisioning	The process that includes all the steps needed in order to fulfil a service subscription request from a user/subscriber. The process may include the following steps: provisioning of information in the end user profile (e.g. traditional settings in the HLR), provisioning of some user information and preferences into a service system, settings in the charging rules for a user, setting for policies to allow request for the user/service, etc.
SLT Instance	A uniquely identifiable service chain that has an accompanying indication for tracing.
SLT Start trigger event	The SLT start trigger event contains criteria that identify the service to be traced and additional criteria such as the time of day that it is to be initiated.
SLT trace indication	An indication that SLT is required and that is included in signalling messages. When a component receives a signalling message that indicates that SLT is required, the component invokes tracing of activities related to that message. The SLT trace indication is passed on to other components in the service chain. An SLT trace token may contain additional information identifying, e.g. the required level of granularity of the logged information.
Trace period	The Period from the marking of a device or component to the unmarking of that device or component.
Trace recording period	Time interval within a Trace Period while trace records are generated for a specific service chain.
User plane	The user plane supports the transfer of user information (e.g. media) between components used to support end-user services.
User profile	See description in [OMADICT].
User provisioning	Provisioning of user data into a system. User provisioning may refer to the registration of the end user into a Service Provider domain (as a new customer), or it may be the provisioning of user information into some service as part of the service subscription provisioning process, etc.

3.3 Abbreviations

CPU	Central Processing Unit
DNS	Directory Name Service
HTTP	HyperText Transport Protocol
IP	Internet Protocol

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

LCM	Life cycle management
MMS	Multimedia Messaging Service
O&M	Operation and Maintenance
OMA	Open Mobile Alliance
OSPE	OMA Service Provider Environment
OSS	Operation Support System
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SLT	Service Level Tracing
SMS	Short Message Service
SNMP	Simple Network Management Protocol
SP	Service Provider
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

4. Introduction (Informative)

4.1 Primary OSPE actors and their motivation

The motivation behind the OSPE may be described from the perspective of the main actors involved with application development, application deployment and application consumption. These actors are identified in the following subsections.

4.1.1 End-User

As described in [OSE] the independent deployment of services leads to inconsistent user experience. From the End-user perspective, the benefits of OSPE are primarily that of improved service experience, i.e. improved services consistency and coherency.

This may simply be in the form of a consistent way in which the end-user interacts with individual services, or the way in which end-users are informed of error situations across multiple services.

By re-using and leveraging components such as a common user-profile it is possible that individual services offered by a Service Provider will avoid, for example, the need for an end-user to repeatedly supply their personal service preferences and settings for each individual service that is offered by the Service Provider.

By standardizing the environment/infrastructure for developing and automating the deployment of services, OSPE will provide the capability for the Service Provider to provide more services, with an improved TTM, to the end-user.

4.1.2 Service Provider

From the Service Provider's perspective, the motivation for the OSPE is three-fold:

- The integration and deployment of services is complicated and expensive, and in many instances the costs involved make some services commercially unfeasible;
- It is difficult and expensive to provide different services to the end-user in a uniform and coherent manner;
- It is very difficult for the Service Provider to quickly and efficiently identify and diagnose faults associated with services that they offer. A Service Provider needs to improve the speed at which they can resolve service failures.

The benefits of the OSPE to the Service Provider therefore include:

- A means to provide different services to the end-user in a uniform and coherent manner, i.e. improved end-user experience;
- Improved service offerings, including time-to-market;
- Reduction in integration efforts for new services;
- Facilitate integrating, deploying, managing, monitoring, upgrading, adding and removing components by standardising the components and identifying their management interfaces. This will also encourage multivendor and hence component plug and play and component reusability;
- It will speed up the resolution of service failures faster and more efficiently whilst reducing operational costs, thus avoiding poor and inconsistent end-user experience.

4.1.3 Application Developer

By defining interfaces into functions like subscriber management, user profile storage, authentication, privacy, etc, Application Developers can invoke such functions in a standard way and therefore reduce the need for services integration efforts when adding a new service enabler into an existing environment. Note that the OSPE Reference Release will not include interface specifications.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

By standardizing the environment/infrastructure for developing and deploying services, OSPE will eliminate the need for the Application Developer to develop proprietary and operator specific mechanisms for the deployment of new services thus increasing the time-to-market for application creation.

By promoting the modularity and re-use of components, the OSPE will in addition allow the Application Developer to speed their time-to-market because they will not need to re-develop those components that are already in use for existing services, but can depend on reusing these functions when necessary. This will inadvertently allow Application Developers to focus their expertise on developing a larger number of richer applications rather than concentrating on infrastructure issues.

4.2 Service life cycle

4.2.1 General

As mentioned in section 4.1 "Primary actors and motivation", the OSPE is focused on improving the service life cycle management (i.e. reduce time-to-market and integration cost). This section defines the service life cycle. Figure 1 shows a simplified view of the service life cycle.

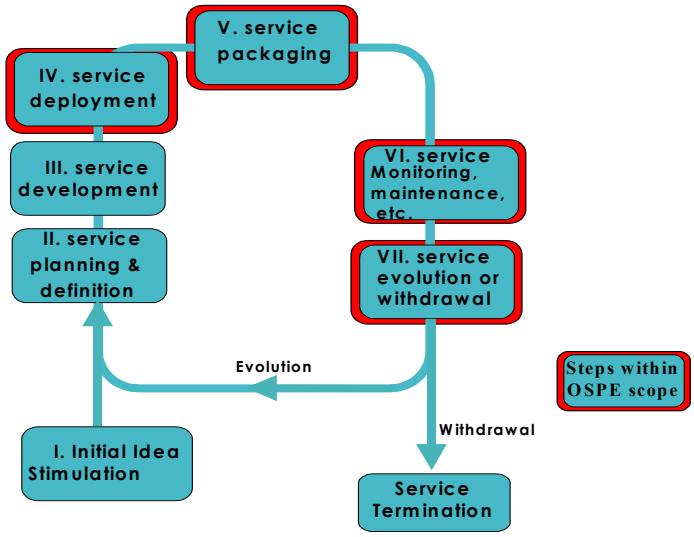

Figure 1: Service life cycle

A brief description of the steps as identified in Figure 1 is as follows:

I. **Initial idea Stimulation**: refers to the process of stimulating the creation of the concepts for an application. (Analysis of market needs, etc, opportunities for new services is identified). When evolving a service to its next release, the idea stimulation process is somehow distributed along the different phase of the original service life cycle, which together with additional thoughts provides experience and ideas for the service evolution.

II. **Service Planning&Definition (Feasibility Study)**: in this phase, after the market, needs, etc. have been analysed, opportunities for new services are identified. If the service is found to be commercially feasible, green light is given to the creation of the service

III. **Service Development:** this phase refers to the process of implementing and testing the applications. Specifications describing the requirements of the new service, design, implementation and tests are normally included in the development of the building components of the new service. Service creation environments with "libraries" are common here, to help in the test process, so that the deployment process on the Service Provider premises goes smoothly henceforth.

IV. **Service Deployment:** refers to the process of deploying the service in the service provider's environment. It includes every step to get the service up and running and ready to be subscribed by end-users (from the service installation to the final tests and service activation)

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

V. **Service Packaging:** This is the process of offering the service to the customer base**.** Commercial packages or bundles are defined here, aiming at concrete user segments, with concrete service features and billing conditions.
VI. **Service Monitoring, Maintenance, etc:** this process involves understanding how well a service is doing (technical and commercial performance, so to speak). Analysis on data gathered from the different service elements is used to know things like trends in service growth, revenue analysis, if the SLA conditions are met, etc. Additionally, some maintenance and management activities on the service will have to be considered in this step.
VII. **Service Evolution or Withdrawal:** the end of a service may be its evolution or its complete termination. Once it is decided that a service offer must be stopped, a clear process must be met in order to properly deal with existing subscriptions, with dependencies among different services, with catalogues, etc.

For OSPE, steps IV, V, VI and VII are applicable. Step III will become also relevant if OMA decides to give support to the development process, defining a service development/creation environment. Step VI may not be fully addressed by OSPE, but it will have to be considered since it may identify further requirements on OSPE activity.

4.2.2 Steps Involved in a Service Deployment

In the context of this document, the service deployment (i.e. every step involved from the service installation to the final tests and service activation) is understood as the steps to be performed by the Service Provider in their Service Provider environment that enable the service to be available and ready to be subscribed to by the Service Provider's end-users.

The following figure shows a simplified description of the steps involved in services deployment.

Figure 2– Service Deployment Steps

These steps as identified in Figure 2 are described below:

1. Service Installation: This covers the physical installation (either hardware or software) of, e.g. servers and platforms. This could imply also the installation of some base software as operating systems, web servers, databases, etc.

2. Service Configuration: A new service will have to declare its dependencies within the Service Provider environment (i.e. the components of the Service Provider environment that will be utilized by the new service). In this process the Service Provider will register the components and functions it will utilize, e.g. location capabilities, messaging, charging. It will also register the need to retrieve user profile information, and that it wishes to be monitored by the Service Provider (i.e. there will dependencies on the Service Provider O&M system). The SLAs associated with each of the components to be used, will be stored.

3. Service Registration: Service Registration forms an important step in the integration process. All information related to the service should be registered, for example, the registration of interfaces such as: the service usage interface (e.g. a URL to call the service), the service subscription provisioning interface (to properly provision user

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

information when subscribing to the service), O&M interfaces in case when the service accepts O&M operations from the Service Provider environment (this is needed for integrity management including fault, configuration and performance management); accounting interfaces (needed if the service is to generate accounting information on demand from the service provider), etc.

4. Service publication: This step is required to publish a service, including all associated attributes, e.g. service type, descriptions, terms and conditions of use, etc. This step is oriented to the service discovery process, which allows end-user's looking for a service, with specific attributes, to find the service in the service catalogue.

5. Service Test: The Service Test process is to ensure that a new service is ready to be subscribed to and consumed by end-users.

6. Service Activation: The forms the last step of the service deployment process. Once the service activated it becomes publicly available and ready for subscription.

All the steps of the service deployment process, with the exception of step 5 and parts of step 1 must be considered in defining an automated process for service deployment in OSPE. Step 5 (Service test) and parts of step 1(Physical installation) will be difficult to automate. However, there will be some aspects of step 1 that will have to be considered, since there will be the need to register all the information regarding any component that may affect the performance of the system.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

5. Use Cases (Informative)

5.1 Use Case A: Deployment of a new application in the Service Provider environment

5.1.1 Short Description

The following use case describes the ability for a Service Provider to reuse existing components of an existing service application called “Games" for the development and deployment of a new Service Provider application called "Restaurant finder" in the same Service Provider environment.

Both of these applications are to be offered to the end-user through the Service Provider's service package called “Entertainment”.

In addition, the Service Provider also identifies the need of an additional component (Location component), which does not already exist in the Service Provider environment, but which is required to be integrated for the support and deployment of the new "Restaurant Finder" application.

This use-case aligns the steps of introducing the new “Restaurant Finder” application with section 4.2. “Service life cycle”.

5.1.2 Actors

Application

- The application may consist of software and/or hardware elements and allow that Service Provider to offer services to their end-users.

Application Developer

- The Application Developer develops the applications and business logic that allows the Service Provider to offer services to their end-users.

End-user

- Consumes the services they have paid for and expects to use the services with consistent and expected quality of service.

Service Provider

- Offers Services to their end-users through their Service Provider environment and guarantees a quality of service for the services that are consumed;
- Manages and administers the Service Provider environment and its associated applications, components and configuration logic. They are also responsible for installing and testing new applications and supporting functions in the Service Provider environment as well as investigating and resolving service related issues, which may be experienced by an end-user.

5.1.2.1 Actor Specific Issues

Application

- The application needs to be developed in a way that reuses existing components and as far as possible avoids duplication and inconsistent data.

Application Developer

- The Application Developer wants to develop applications as fast as possible, and reuse its applications in as many Service Provider environments as possible. The Application Developer needs to be aware of existing applications, business logic and what reusable components are available for reuse in the Service Provider environment.

End-user

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

- Consumes the services they have paid for and expects to use the services with consistent and expected level of quality, service usability and user experience. The deployment of new applications and components should not impact existing live services.

Service Provider

- The Service Provider wants to deploy services in its Service Provider environment in a fast and efficient manner, using automated processes as far as possible, and avoiding expensive integration costs. The Service Provider needs to understand the existing applications and which components are being reused across applications. They also need to ensure that when a new application or supporting component is deployed in the Service Provider environment existing services are not impacted.

5.1.2.2 Actor Specific Benefits

Application

- The speed at which viable applications are developed and deployed will be improved.

Application Developer

- The Application Developer will be able to concentrate development effort on the unique aspects of the new application and will be able to reuse existing components and data. This will improve the time for new services being made available to end-users. It will also help avoid duplication of functionality and data across different enabler hence reducing complexity of application logic. Integration efforts for allowing applications to run in the Service Provider environment will be limited and the applications will be more reusable across different Service Providers.

End-user

- End-users will be able to consume wider range of new services and will experience no degradation in existing services during the installation of the new service.

Service Provider

- The Service Provider is able to deploy a wider range of new services quickly and efficiently, with reduced complexity and integration efforts, whilst maintaining service quality, which also includes Service continuity;
- The administration and management the Service Provider environment is simplified through the introduction of standard life cycle management processes, for example, the integration processes for applications and components in the Service Provider environment will be simplified;
- The Service Provider better understands the capabilities and the relationship between components and applications. Through the avoidance of duplicated data, it will also improve the engineer's ability to resolve end-user subscription/profile related issues across multiple services.

5.1.3 Pre-conditions

- The Service Provider has completed the Idea Stimulation and Service Planning phases (as described in section 4.2 "Service life cycle".
- The Service Provider has decided and scoped how the Service will be packaged (see Service Packaging in section 4.2 "Service life cycle". This phase includes but not limited to:
    - The Service Provider determines which of their customer segment (e.g. consumer or corporate) will be offered the new application;
    - The Service Provider determines how to "bundle" the new application in a service offer, e.g. does the application form part of a standalone service or is the application packaged with other applications to for a Service Provider's promotion;
    - The Service Provider determines the appropriate charging and billing plan for that particular customer segment and arranges the billing conditions.
- The Service Provider has completed the Service Development phases (as described in section 4.2 "Service life cycle". This phase includes but not limited to:

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

  - The Application Developer identifies the existing components in the Service Provider environment needed to support (or provide a service to) the new application;
  - In addition, the Service Provider identifies the need for the integration of a new component "Location" and the upgrade of other components in order to support the new "Restaurant Finder" service;
  - The Service Provider purchases the new component according to the new application requirements.

- A Service Provider environment exists, which has reusable components and supports multiple applications including the "Games" application.
- The Service Provider environment provides information as to what components (including their settings and connectivity to other components) are in the environment. Information related to the characteristics of each component is also made available by the Service Provider environment.
- A number of the components in the Service Provider environment are potentially reusable across the "Games" application and the "Restaurant finder" application.
- End-user has an account with the Service Provider.
- The Service Provider has defined the requirements for the new "Restaurant finder" application and has identified the need for the installation and deployment of a new "Location" component.
- The Service Provider has generates the relevant information in order to install and deploy the component in the Service Provider environment.
- The Application Developer, through interfaces exposed by the Service Provider environment is able to determine what components are available for use in the Service Provider environment.
- There are a number of live Service Provider services that the Service Provider's end-users are consuming.
- The Service Provider environment and its components expose life cycle interfaces that enable the ability to perform life cycle functions in order to deploy the new application.
- A number of functions exist for the purpose of supporting the Service Provider environment:
  - A Catalogue that holds the details (e.g. names and characteristics) of all the deployed components;
  - A Catalogue that holds the details of all the Service Provider services;
  - A Catalogue that describes the relationship and dependencies between the components and the applications;
  - A mechanism (e.g. self-care application) that provides the ability for end-users to subscribe to Service Provider services;
  - O&M functionality (e.g. Service Level Tracing, Alarms and Statistical analyzers);
  - Management function for User profile information.

5.1.4 Post-conditions

- The new component is integrated into the Service Provider environment and a number of existing components are upgraded in order to support the requirements of the "Restaurant Finder" application.
- The new "Restaurant Finder" application logic is deployed in the environment.
- The Service Provider makes the new "Restaurant Finder" application available through its Entertainment service to its end-users.
- The end-user continues to consume their existing services and is now able to consume the new "Restaurant Finder" application, getting a consistent and expected end-user experience each time.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

- The upgraded component continues to fully support the previously existing applications, i.e. there are no backward compatibility problems.
- In general, the successful integration of the application can be observed by:
  - The details of the new "Location" component are correctly incorporated in the component catalogue;
  - The details of the new application are correctly incorporated in the Services Catalogue, and the application is appropriately categorised according to the end-user segment that will be offered the service;
  - The end-user may subscribe to, customise and unsubscribe from the new application using the normal service Provider's self care application;
  - Policies may be applied for requests to the new service;
  - Authorization, authentication and charging functions are aware of the new application, and are prepared to receive requests related to the consumption of this application.

5.1.5 Normal Flow

Life cycle phase: Service Deployment

1. The Application Developer performs a pre-test of the application in their environment before installing into the Service Provider's environment. This allows the Application Developer to guarantee a quality of service.
2. Using the automated processes of OSPE (e.g. life cycle interfaces and capabilities), the Service Provider:
   1) Installs the new "Restaurant Finder" application in the Service Provider environment;
   2) Installs the new "Location" component application in the Service Provider environment;
   3) Installs the configuration logic for the new Application (e.g. configuration logic to define how the application runs in the Service Provider environment);
   4) Upgrades the existing components in the Service Provider environment (This may include configuring component settings as well as pointers (e.g. addresses, URI) to other components (new and old);
   5) Includes the details of the new "Restaurant Finder" application in the Service Providers catalogue;
   6) Includes the details of the new "Location" component in the components catalogue and also revises the information associated to the upgraded existing components;
   7) Updates the catalogue that describes the relationship and dependencies between the new "Restaurant Finder" application and the new and existing components;
   8) Performs a back-up of all settings and data associated to each of the components;
   9) Associates the new application and components with the existing O&M functions;
3. The new application is assigned to the correct service package as defined by the Service Provider.
4. When the new application, component (including upgraded components) and configuration logic is installed, the Service Provider performs pre-live testing (see use-cases for "Service Level Tracing").
5. When the tests are successfully approved, the Service Provider publishes the new "Restaurant Finder" application (i.e. the application becomes active and end-user start to consume the application).
6. The Service Provider tests the new live application and other live services to ensure that the deployment of the new application and its associated configuration does not impact existing live services (see use-case "Regression testing following network fix").

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

5.1.6 Alternative Flow

5.1.6.1 Component upgrades

Normal flow, bullet#4: It is possible that the old components are not the appropriate version to support the new application. In such a case, the Service Provider must determine the impact of performing and upgrade of the old component.

If the new version of the component is backward compatible, then the normal flow applies. However, if there are incompatibilities between applications dependent on old component versions, the Service Provider must decide between upgrading the older incompatible applications or maintaining within the Service Provider environment the old version of the problematic component. In the latter case, calls to the component must be appropriately re-directed to the appropriate version.

When installing new applications, the re-direction rules must be updated based on the requirements of these new applications.

5.1.6.2 Setting changes

Normal flow 2, bullet#4: It is possible that the settings of a component must change in order to support the new application. The Service Provider must determine if two versions of the components should be maintained within the environment or if it is possible dynamically change the settings of the component depending on the application that calls it.

5.1.6.3 Data sub-setting

Normal flow 2, bullet#4: It is possible that the data used by a component must change in order to support the new application. This can be considered as a particular case of the settings discussed in section 5.1.6.2, where a setting is used to select the appropriate data sub-set.

5.1.7 Operational and Quality of Experience Requirements

Alternative Flow: Operational aspects: Setting changes.

Dynamic data settings may be performed by an additional component that monitors the type of call and configures the components on the fly in a side channel. It is also possible that the component offers an interface to the Service Provider that enables dynamic configuration.

Intercepted calls to the components can be composed (the intermediary performs the necessary configuration steps) or transformed (into a new call that is the original call and configuration call) by the intermediary that intercepted the call.

During operation, the Service Provider can tune the ratio of components with one setting versus the other based on usage. This can even be dynamically achieved using optimisation algorithms.

5.2 Use Case B: Removal of an application from the Service Provider environment

5.2.1 Short Description

The following use case describes the ability for a Service Provider to remove an existing application called “Games” and associated components from the Service Provider environment.

5.2.2 Actors

5.1.2.1 Roles

Application

- The application may consist of software and/or hardware elements and allow that Service Provider to offer services to their end-users.

End-user

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

- Consumes the services they have paid for and expects to use the services with consistent and expected quality of service.

Service Provider

- Offers services to their end-users through their Service Provider environment and guarantees a quality of service for the services that are consumed.
- Manages and administers the Service Provider environment and its associated applications, components and configuration logic. They are also responsible for installing and testing new applications and supporting functions in the Service Provider environment as well as investigating and resolving service related issues, which may be experienced by an end-user.

5.1.2.2 Actor Specific issues

Application

- The application needs to be removed from the Service Provider environment without affecting other live services.

End-user

- Consumes the services they have paid for and expects to use the services with consistent and expected level of quality, service usability and end-user experience. The removal of applications and components should not impact existing live services.

Service Provider

- The Service Provider wants to remove services from their Service Provider environment in a fast and efficient manner, using automated processes as far as possible, and avoiding expensive decommissioning costs. The Service Provider needs to understand the existing applications and which components are being reused across applications. They also need to ensure that when an application or supporting component is removed from the Service Provider environment existing services are not impacted;
- The Service Provider needs to ensure all active subscriptions to the service are deactivated before the service is removed from the Service Provider environment (e.g. to ensure service subscription consistency and general good end-user experience (as stated above)), and that billing conditions are adjusted accordingly.

5.1.2.3 Actor Specific benefits

Application

- The speed at which obsolete applications are decommissioned and removed from the Service Provider environment will be improved.

End-user

- End-users will experience no degradation in existing services during and after the removal of the obsolete service.

Service Provider

- The Service Provider is able to decommission and remove obsolete services quickly and efficiently, with reduced complexity and efforts, whilst maintaining service quality, which also includes service continuity.

5.2.3 Pre-conditions

- The Service Provider has defined the business criteria for the removal of the “Games” application from the Service Provider environment.
- The Service Provider has also identified the need to remove a redundant component (that was used to support the “Games” application from the Service Provider environment and has identified no reasons to perform the downgrade of other existing components.
- A Service Provider environment exists, which has reusable components and supports multiple live applications including the “Games” application, which is being consumed by the end-users.
- The Service Provider environment provides information as to what components are integrated in the environment. Information related to the characteristics of each component is also made available by the environment.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

- A number of the integrated components are being reused across the "Games" application and other live services.
- End-user has an account with the Service Provider.
- The Service Provider environment and its components expose life cycle interfaces that enable the ability to perform life cycle functions in order to remove the obsolete "Games" application.
- The application to be withdrawn exists in a service package (see section 4.2 "Service life cycle") that is assigned for a specific customer segment (e.g. consumer or corporate) and is associated with a specific charging and billing conditions.
- There are no active end-user subscriptions for the service that is to be withdrawn.
- No other services/applications are using the application that is to be removed.
- The Service Provider has determined the impacts on the existing billing conditions associated with the Service Package from which the service is to withdrawn.
- A number of functions exist for the purpose of supporting the Service Provider environment:
    - A catalogue that holds the details (e.g. names and characteristics) of all the deployed components (e.g. those components used to support the "Games" application);
    - A catalogue that holds the details of all the Service Provider services (e.g. including "Games" service);
    - A catalogue that describes the relationship and dependencies between the components and the applications;
    - O&M functionality (e.g. Service Level Tracing, Alarms and Statistical analyzers);
    - Management function for user profile information.

5.2.4 Post-conditions

- The end-user continues to consume their remaining services getting a consistent and expected end-user experience each time but is unable to consume the obsolete "Games" application.
- The "games" application and the obsolete component and all associated application logic is removed from the Service Provider environment.
- In general, the successful removal of the application can be observed by:
    - The details of the obsolete component are successfully removed from the component catalogue;
    - The details of the "Games" application is successfully removed from the services catalogue;
    - The end-user is unable to subscribe to, customise and unsubscribe from the removed "Games" application;
    - All Service Provider environment resources that were dedicated to the "Games" application are relinquished and made available for other applications, if appropriate;
    - The service is "detached" from the service package(s);
    - If applicable, the billing conditions associated with the service package are correctly adjusted;
    - The authorization and charging functions are aware of the withdrawal of the application, and will reject requests related to the consumption of the service.

5.2.5 Normal Flow

Life cycle phase: Service Withdrawal

1. Using the automated processes of OSPE (e.g. life cycle interfaces and capabilities), the Service Provider:

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

1) Checks that there are no active subscriptions to the application;
2) Checks that there are no other services/applications using the application to be removed;
3) Removes the obsolete "Games" application from the Service Provider environment;
4) Removes the obsolete configuration logic associated with the "Games" Application (e.g. configuration logic used to define how the application runs in the Service Provider environment);
5) Removes the details of the obsolete "Games" application from the Service Providers catalogue;
6) Detaches the obsolete Service from its Service Package(s);
7) For each component used by the application, if no other application/component needs this component, then:
    1. Removes the obsolete components from the Service Provider environment;
    2. Removes the details of the obsolete component from the components catalogue.
8) Updates the catalogue that describes the relationship and dependencies between the application and existing components to illustrate:
    1. "Games" application has been removed from the Service Provider environment;
    2. Component has been removed from the Service Provider environment;
    3. The removal of dependencies between existing components and the obsolete application;
    4. The deallocation of other resources in the Service Provider environment and as far as possible to return the configuration of the Service Provider environment to that state prior to the installation of the "Games" application;
    5. Disassociates the obsolete "Games" application and component from the existing O&M functions.

2. When the obsolete "Games" application, component and configuration logic is removed, the Service Provider performs pre-live testing (see use-cases for "Service Level Tracing").

5.2.6 Alternative Flow

As described in "Post conditions"

5.3 Use Case C: Fault detection and repair of components

5.3.1 Short Description

This use case describes some of the steps involved in monitoring the components in a Service Provider environment and predicting, detecting or repairing a fault at the component level.

5.3.2 Actors

The Service Provider

- Manages the applications, services and components within the Service Provider environment.

5.3.2.1 Actor Specific Issues

Service Provider

- Maintain every existing service in good working condition.
- Manage and administer components at any stage of their life cycle. In particular, while in production (i.e. deployed):
    - Monitor usage, load and behaviour of the components;
    - Detecting problematic behaviours;
    - Predicting problems;

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

- Detecting Faults;
- Replacing a component (when faulty or when at risk).

5.3.2.2 Actor Specific Benefits

Service Provider

- Being able to manage throughout their life cycle and administer each component.
- Being able to avoid or reduce down time of the service that the component supports.
- Being able to predict trouble for internal purpose as well as possibly to offer a service of early warning of premium customers.

5.3.3 Pre-conditions

- A Service Provider environment exists and it consists of several components.
- Services are being consumed by the Service Provider's end-users, and these services are supported by the components in the Service Provider environment.
- The Service Provider environment is managed by the Service Provider through an extensible application that relies on the life cycle management (including monitoring) interfaces of each component.
- The Service Provider environment is periodically backed up.

5.3.4 Post-conditions

- Problems have been forecasted or detected and the associated issues have been fixed.
- The overall Service Provider environment is operational for the current and foreseen usage levels.

5.3.5 Normal Flow

1. The Service Provider monitors the different components (including services) in its Service Provider environment.
2. The Service Provider administration application interrogates (polling) the different components in the system to, for example, query about:
    1) Error log:
        I. Application level (e.g. exception thrown in answer to a request).
        II. Internal error message (i.e. not exposed to the requestor).
    2) Load:
        I. Amount of transactions or processed requests.
        II. Number of instances created for the component.
    3) Time per request:
        I. Time for a component to process a request.
        II. Delay experiences by other components that made requests.
    4) Self diagnosis results:
        I. Performed on request, or
        II. Last results.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

3. The Service Provider inspects the results of the request (e.g. visually or it is performed through automated processes via a monitoring application).

4. The Service Provider detects that one component has a problematic behaviour. Simple examples of this could come from:

    1) Detecting that a requestor experiences an ever-increasing delay with that component while no noticeable network congestion is observed;

    2) Detecting that a component is overloaded (many requests, many instances, increased delays);

    3) Increased amount of exceptions and internal error message that could denote for example corruption of a data file or upgrade failure / integration problem with respect to another component (or a new service);

    4) Problematic self diagnosis messages;

    5) Internal error messages;

    6) Requests always fail (e.g. misconfigured component that does not know the correct address for the target to its request).

5. The Service Provider may decide to broadcast a warning message to appropriate parties (internal or customers).

6. The Service Provider decides to take corrective / pre-emptive actions by either:

    1) Replacing the component;

    2) Duplicating the component (e.g. in case of overload);

    3) Allocating more underlying resources to the component: Upgrading a component or performing a change of settings (e.g. if there are some incompatibilities that lead to errors when some services call the component).

7. Once corrective / pre-emptive action has been taken by the Service Provider the normal Service Provider operation and administration can continue.

5.3.6 Alternative Flow

It is possible that the detection phase does not lead to early diagnosis of a problem.

Instead, the following case could be met.

5.3.6.1 Detection of a component fault

The Service Provider administration/monitoring application is registered as a listener for a possible error message thrown by different components.

A particular fault in the Service Provider environment may trigger such an event.

The monitoring application catches the event and launches the polling step 2 in section 5.3.5 and the flow follows as in that section. As there may be a system failure (e.g. component not responding, fatal error message), immediate action must be taken and the application may send appropriate alerts to the Service Provider.

5.3.7 Operational and Quality of Experience Requirements

Normal flow #6: Operational aspects of corrective/pre-emptive actions:

Replacing the component: Component settings are transferred (assumed correct)

All requests from other components are re-directed:

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

- Via address redirection: If the Service Provider environment uses such an addressing scheme, e.g. DNS or allows request routing; or
- By updating all the settings of all the other components that rely on it: The Service Provider must be able to visualize / track all the dependencies between components and update the settings on the fly.

Duplicating the component (e.g. in case of overload)

- Component settings are mirrored (assumed correct);
- Request are balanced between the components.

Allocating more underlying resources to the component:

- The Service Provider configures the component to take advantage of the additional underlying resources (e.g. CPU, Hardware, memory etc…). For example the maximum amount of instances that can be created is increased;
- Upgrading a component or performing a change of settings (e.g. if there are incompatibilities that lead to errors when some services call the component) (In addition see section 5.1.6.1 Component upgrades.

5.4 Use Case D: Integrating a new component in a Service Provider environment

5.4.1 Short Description

This use case describes some of the integration steps involved with integrating a component in a Service Provider environment.

5.4.2 Actors

Service Provider

- Administers the applications services and components within the Service Provider environment.
- Responsible for integrating a component within the Service Provider environment.

5.4.2.1 Actor Specific Issues

Service Provider

- Maintain every existing service in good working condition.
- Manage and administers the existing services at any stage of their life cycle.
- Integrates a component within the service provider domain.

5.4.2.2 Actor Specific Benefits

Service Provider

- Being able to manage throughout their life cycle and administer each component (and therefore services).
- Being able to integrate a component in the Service Provider environment.

5.4.3 Pre-conditions

- A Service Provider environment exists and it consists of several components.
- Services are deployed, supported by these components.
- The Service Provider environment is managed by the administrator through an extensible application that relies on the life cycle management (including monitoring) interfaces of each component.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

- A new component must be deployed in the Service Provider environment.
- There are adequate resources in the network to support the new service.

5.4.4 Post-conditions

Service Provider

- The new component is integrated within the Service Provider environment.
- Services can rely on the components.
- The Service Provider can administer the components.

5.4.5 Normal Flow

1. The Service Provider identifies the dependencies that the new component has on other components.
2. The new component is deployed in the Service Provider environment by establishing the necessary physical connections if required.
3. The new component is configured through the life cycle management interfaces that it exposes. This may include configuring the component settings as well as pointers (e.g. addresses, URI,) to other components (new and older) that it depends on.
4. Documentation of the version, settings, interfaces and dependencies is updated (starting from the manufacturer details) and appropriately stored.
    1) Application interfaces may be registered in a registry for discovery.
    2) Other information may be stored elsewhere and be available for future administration or integration tasks.
5. The Service Provider extends the administration application to the new component by integrating the modules provided by the vendor of the new components.
6. The Service Provider can manage all aspects of the life cycle of the new component, older services and components (new and older).

5.4.6 Alternative Flow

None identified for this use case.

5.4.7 Operational and Quality of Experience Requirements

When developing a service, it is good practice to expose similar interfaces (versioning, configuration and monitoring) to components in general.

5.5 Use Case E: Automated testing of Service Provider services

5.5.1 Short Description

This use case describes the scenario when a Service Provider performs periodic automated testing of their key live services. After one particular set of periodic batch tests a service error is identified with one of the services. On the identification of the service error, the Service Provider retrieves that logged information and fixes the cause of the fault.

5.5.2 Actors

End-user
- The end-users who use a mobile device to consume a service provided by their Service Provider.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

Service Provider
- Supports the necessary network infrastructure to support and provide services to their customers. The Service Provider may also integrate Services from a third party Service Provider to enhance their services portfolio;
- They are the end-user's first point of contact for any service related queries, e.g. subscription, payment and billing related, and service specific queries or issues;
- Manages and maintains the day-to-day operations of the Service Provider's network and supporting services;
- Designs, develops and implements the service logic that may be realized as a complex set of interactions between enablers deployed by both the Service Provider and externally by a third party Service Provider.

5.5.2.1 Actor Specific Issues

End-user
- Consumes the services they have paid for and expects to use the services with consistent and expected level of quality.

Service Provider
- Offers Services to their end-users and guarantees a quality of service for the services that are consumed;
- The primary contact point for their end-user. They need to ensure that the customer's complaints are dealt with in an effective and efficient manner to maintain customer satisfaction;
- Ensure it has all means/tools to be able to manage the services and solve/address possible issues an end-user may experience or which may impact end-user experience.
- Ensures that their service designs meet the expected market needs in terms of time-to-market and quality of service, and end-user expected quality of service, e.g. service works first time and is simple use etc;
- They need to able to make the necessary corrections to the service logic whenever a service fault is detected.

5.5.2.2 Actor Specific Benefits

End-user

- Consistent and expected level of quality with every service execution.

Service Provider

- Able to maintain and improve the quality of service offered to their end-users;
- Maintains and improves level of customer satisfaction through efficient and effective resolution of end-user complaints. It will help in providing more quality fault related information to appropriate support teams within the Service Provider when a fault occurs;
- Confidence that issues can be tracked, isolated and corrected in an effective and efficient manner. This enables the process of fixing and debugging the issues more quickly. It will also help in providing more quality fault related information not only to other appropriate support teams within the Service Provider but to third party Service Providers when a fault occurs.

5.5.3 Pre-conditions

- End-user has an account with the Service Provider and their subscription is fully paid.
- There are a number of live Service Provider services that the Service Provider's end-users are consuming.
- A Services test emulator is connected into the live network and has been configured by the Service Provider to invoke pre-scripted test routines every 4 hours on a daily basis.
- The pre-scripted test routines simulates the end-user activity for initiating and consuming a specified set of live services, which when initiated replicate the same call/session set-up and completion phase as initiated by the end-user.
- The Service Provider's network infrastructure, including their test emulator, supports the Service Level Tracing functionality.
- The Service Provider, through, for example, a central operations console, is able to connect to and retrieve logged service level diagnostic information from every node in the Service Provider's network.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

5.5.4 Post-conditions

- The test emulator has invoked the pre-scripted test routines and the test emulator has raised an error alarm when one of the services fails to work correctly.
- The Service Provider has responded to the error alarm and have retrieved from the network all the logged information associated with the specific pre-scripted test routine.
- The cause of the error alarm and the identified causes of service failure are rectified by the Service Provider.
- The end-user continues to consume their services getting a consistent and expected end-user experience each time.

5.5.5 Normal Flow

1. The Service Provider's test emulator initiates a scheduled set of pre-scripted service tests, which invokes the Service Providers key services, e.g. Send MMS, streaming, payment, etc. Each pre-scripted service test also initiates service level tracing for each service that it invokes.
2. During the test phase the test emulator raises an error alarm when it identifies that that one of the test cases has failed.
3. The Service Provider identifies the failed test case, and makes some basic network checks, e.g. no obvious signs of network node failure. However, they are unable to determine through basic checks the cause of the failure.
4. The Service Provider retrieves all logged service level trace information from all nodes in the Service Provider network.
5. At this point in time the Service Provider decides to inform the appropriate support teams within the Service Provider that there may be a service fault with a particular live service and that the cause of failure is being investigated.
6. The Service Provider determines that the cause of fault is related to an enabler component and creates a fault report.
7. The Service Provider identifies the cause of the fault, e.g. they identify the faulty service component in the network, and are able to perform an immediate node fix.
8. The fault is corrected and the Service Provider repeats an unscheduled emulator test (could be classed as a Regression test).
9. The result of the unscheduled emulator test confirms that the fault diagnoses and associated service component fix is successful. The Service Provider then informs the appropriate support teams within the Service Provider that the Service has now resumed normal service.

5.5.6 Alternative Flow

5.5.6.1 Alternative Flow 1: The Service Provider is unable to fix the fault

1. As per normal flow 1 to 5.
2. The Service Provider is unable to identify the cause of the fault and the logged service level trace information is further analysed.
3. The Service Provider analyses the information and identifies the cause of fault, e.g. the cause of failure may be related to a previous network correction for rectifying a different service failure.
4. The Service Provider are unable to quickly rectify the cause of the service failure (e.g. a major software release required) and therefore works towards identifying a short term fix to the network.
5. The Service Provider identify a short-term fix, which is implemented.
6. As per normal flow 8.
7. The result of the unscheduled emulator test confirms that the short-term fix is successful. The Service Provider informs the appropriate support teams that the Service has now resumed normal service.
8. The Service Provider continues to develop a long-term fix for the problem.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

5.5.7 Operational and Quality of Experience Requirements

The entire use-case covers operational requirements.

5.6 Use Case F: Regression testing following a network fix

5.6.1 Short Description

This use cases describes the scenario when a Service Provider initiates network regression testing of the existing live services following a change to the Service Provider's network infrastructure, e.g. to introduce a new service, or new network node, or a software patch for an existing service.

After performing the set of specified regression tests it is possible for the Service Provider to quickly and efficiently identify whether the network changes have impacted any of the live services, before any of the Service Provider's end-users experience a service fault.

If there are any services that have been impacted then the Service Provider is able to rollback the change to avoid any further disruption to the live services, and allowing time to identify and possibly fix the associated network fault.

5.6.2 Actors

End-user
- The end-users who use a mobile device to consume a service provided by their Service Provider.

Service Provider:
- Supports the necessary network infrastructure to support and provide services to their customers. The Service Provider may also integrate Services from a third party Service Provider to enhance their services portfolio;
- They are the end-user's first point of contact for any service related queries, e.g. subscription, payment and billing related, and service specific queries or issues;
- Manages and maintains the day-to-day operations of the Service Provider's network and supporting services;
- Designs, develops and implements the service logic that may be realized as a complex set of interactions between enablers deployed by both the Service Provider and externally by a third party Service Provider.

5.6.2.1 Actor Specific Issues

End-user
- Consumes the services they have paid for and expects to use the services with consistent and expected level of quality.

Service Provider
- Offers Services to their end-users and guarantees a quality of service for the services that are consumed;
- The primary contact point for the end-user. They need to ensure that the customer's complaints are dealt with in an effective and efficient manner to maintain customer satisfaction;
- Ensure it has all means/tools to be able to manage the services and solve/address possible issues an end-user may experience or which may impact end-user experience;
- Ensures that their service designs meet the expected market needs in terms of time-to-market and quality of service, and end-user expected quality of service, e.g. service works first time and is simple use etc;
- They need to able to make the necessary corrections to the service logic whenever a service fault is detected.

5.6.2.2 Actor Specific Benefits

End-user

- Consistent and expected level of quality with every service execution.

Service Provider

- Able to maintain and improve the quality of service offered to their end-users;

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

- Maintains and improves level of customer satisfaction through efficient and effective resolution of end-user complaints. It will help in providing more quality fault related information to the appropriate support teams within the Service Provider when a fault occurs;
- Confidence that issues can be tracked, isolated and corrected in an effective and efficient manner. This enables the process of fixing and debugging the issues more quickly. It will also help in providing more quality fault related information not only to other appropriate support teams within the Service Provider but to third party Service Providers when a fault occurs.

5.6.3 Pre-conditions

- End-user has an account with the Service Provider and their subscription is fully paid.
- There are a number of live Service Provider services, including the service that is to be modified, that their end-user customers are consuming.
- The Service Provider's network infrastructure, including their test emulator, supports the Service Level Tracing functionality.
- A Services test emulator is connected into the live network and has been configured by the Service Provider to invoke pre-scripted regression test routines.
- A business need has been identified for the development and network wide rollout of a service logic modification.
- The Service Provider is able to connect to and retrieve logged service level diagnostic information from every node in the Service Provider's network.
- The Service Provider has pre-tested the service logic modification on a non-live network configuration and has released/signed-off the fix along with the scripted file that allows the Service Provider to load the fix on all appropriated nodes in the network.

5.6.4 Post-conditions

- The Service Provider has invoked the pre-scripted test routines and the test emulator has raised an error alarm when one of the services fails to work correctly.
- The Service Provider performs a rollback of the new service logic modification and repeats the regression tests to ensure that all existing live services are working correctly.
- The Service Provider retrieves the logged service trace information, identifies the cause of fault, and the fault is rectified.
- The end-user continues to consume their services getting a consistent and expected end-user experience each time.

5.6.5 Normal Flow

1. The Service Provider develops the service logic modification and the appropriate run-time scripts, which enables the Service Provider to load the modification on all appropriate nodes.
2. The Service Provider initiates the run-time scripts on all nodes to install the service logic modification.
3. The Service Provider invokes their standard pre-scripted regression tests from their test emulator, which invokes the Service Provider's key services, e.g. Send MMS, streaming session etc.
4. During the regression testing the test emulator raises an alarm when it determines that a specific test case has failed.
5. The Service Provider identifies the failed test case, and makes some basic network checks, e.g. have any of the appropriate nodes in the network failed the activation stage. However, they are unable to determine through basic checks the cause of fault.
6. At this point in time the Service Provider makes the decision to restore the network configuration to it original state to ensure that the existing live services are not impacted.
7. The Service Provider retrieves (e.g. from their remote console) all logged trace information associated with the failed test.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

8. The Service Provider analyses the retrieved logged service trace information and determines the cause of the service failure, e.g. the service logic modification is not compatible with on of the software builds of an individual node.
9. The Service Provider creates a detailed fault report, based on the retrieved logged trace information.
10. Following the rollback of the service logic modification the Service Provider again invokes their standard pre-scripted regression tests to ensure that the existing live Services continue to work correctly.
11. The Service Provider corrects the error in the service logic modification and the Service Provider rolls out the service logic modification at a later date.

5.6.6 Alternative Flow

5.6.6.1 Alternative Flow 1: The Service Provider fixes the fault

1. As per normal flow 1 to 5.
2. Considering the very low traffic levels and hence the limited risk, the Service Provider decides not to roll-back the service logic modification in order to restore the network configuration to it original state.
3. As per normal flow 9.
4. The Service Provider analyses the retrieved logged service trace information and determines the cause of the service failure, e.g. there is identifiable error in the service logic modification.
5. As per normal flow 7.
6. The Service Provider corrects the identifiable error in the service logic modification and re-creates a set of run-time scripts.
7. As per normal flow 2 and 3.
8. No errors are raised at the operations test emulator.

5.6.7 Operational and Quality of Experience Requirements

The entire use-case covers operational requirements.

5.7 Use Case G: Service fault localization within a Service Provider's network

5.7.1 Short Description

This use case describes the use of a mechanism to assist end-to-end service investigation by a Service Provider.

The main aim of the use case is to describe how the mechanism is used to identify the location of a fault in the Service Provider's network that supports a set of service enablers that provide an end-user Service. The end-user Service is invoked, the service is traced though the service enabler realisations, and the Service Provider analyses the Service Trace to find the location of the fault.

5.7.2 Actors

End-user
- The end-users who use a mobile device to consume a service provided by their Service Provider.

Service Provider
- Supports the necessary network infrastructure to support and provide services to their customers. The Service Provider may also integrate Services from a third party Service Provider to enhance their services portfolio;
- They are their end-user's first point of contact for any service related queries, e.g. subscription, payment and billing related, and service specific queries or issues;
- Manages and maintains the day-to-day operations of the Service Provider's network and supporting services;

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

- Designs, develops and implements the service logic that may be realized as a complex set of interactions between enablers deployed by the Service Provider.

5.7.2.1 Actor Specific Issues

End-user
- Consumes the services they have paid for and expects to use the services with consistent and expected level of quality.

Service Provider
- Offers Services to their end-users and guarantees a quality of service for the services that are consumed.
- The primary contact point for the end-user. They need to ensure that the customer's complaints are dealt with in an effective and efficient manner to maintain customer satisfaction.
- Ensure it has all means/tools to be able to manage the services and solve/address possible issues an end-user may experience or which may impact end-user experience.
- Ensures that their service designs meet the expected market needs in terms of time-to-market and quality of service, and end-user expected quality or service, e.g. service works first time and is simple use etc.
- They need to able to make the necessary corrections to the service logic whenever a service fault is detected.

5.7.2.2 Actor Specific Benefits

End-user

- Consumes the services they have paid for and get a consistent and expected level of quality with every service execution.

Service Provider

- Able to maintain and improve the quality of service offered to their end-users;
- Maintains and improves level of customer satisfaction through efficient and effective resolution of end-user complaints. It will help in providing more quality fault related information to the appropriate support teams within the Service Provider when a fault occurs;
- Confidence that issues can be tracked, isolated and corrected in an effective and efficient manner. This enables the process of fixing and debugging the issues more quickly. It will also help in providing more quality fault related information not only to other appropriate support teams within the Service Provider but to third party Service Providers.

5.7.3 Pre-conditions

- End-user has an account with the Service Provider and their subscription is fully paid.
- There are a number of live Service Provider services that their end-user customers are consuming.
- The Service Provider's network infrastructure supports the Service Level Tracing functionality. This also means that when a device is marked by the Service Provider, any calls/sessions invoked (until the Service Provider unmarks the device) from that device initiates service level tracing.
- The Service Provider have the right type of mobile devices that are capable of replicating the service failure.

5.7.4 Post-conditions

- The Service Provider replicates the service failure.
- The Service Provider retrieves the logged service trace information and identifies the fault that causes the failure
- The cause of the service failure is identified as being part of the Service logic within the Service Provider's network. The fault that causes the service failure is rectified.
- The end-user continues to consume their services getting a consistent and expected user experience each time.

5.7.5 Normal Flow

1. A number of Service Provider end-users experience a fault with a service they have paid for, and after a number of failed attempts contact their Service Provider.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

2. The Service Provider performs a number of basic checks (e.g. checks end-user has enough money/air time credit) but is unable to identify the cause of error.
3. The Service Provider checks the network for error alarms and examine the statistical information associated with the key network nodes for signs of unusual nodal behaviour. However, they are unable to identify the cause of failure.
4. The Service Provider checks the service configuration and identifies that a fault may reside in the Service Provide domain, as they are the providers of the application.
5. The Service Provider does not ask the end-user to replicate the fault as the Service Provider is able to replicate the failure from their test mobile phone.
6. The Service Provider marks their test mobile phone and then replicates the failure case.
7. Each node (e.g. component) in the Service Provider network logs all associated trace information.
8. The Service Provider retrieves the associated logged service level tracing information from all nodes in their network.
9. The Service Provider analyses the retrieved trace information for the cause of the fault.
10. The fault is identified and located within the Service Provider's network.
11. After the fault is fixed the Service Provider repeats the test and confirms that the service is now working correctly.
12. The Service Provider contacts the end-users to inform them that the service is now working correctly.

5.7.6 Alternative Flow

5.7.6.1 Alternative Flow 1: Service Provider cannot replicate service failure but uses other means to identify cause of failure

1. As per normal flow 1 to 4.
2. The Service Provider is unable to replicate the service fault with their engineering test mobile phone.
3. The Service Provider chooses other means for identifying the cause of service failure, however this takes a considerable amount of time and effort across the various support teams.
4. As per normal flow 11 and 12.

5.7.7 Operational and Quality of Experience Requirements

The entire use-case covers operational requirements.

5.8 Use Case H: Service fault localization when a service is hosted by a third party Service Provider

5.8.1 Short Description

This use cases describes the scenario when a number of customers complain to their Service Provider about a fault that they repeatedly experience when trying to access a particular application that is offered by the Service Provider.

After performing some basic service checks the Service Provider confirms that a third party Service Provider hosts the application on their behalf and that there is a need to identify whether the fault resides either in the Service Provider domain or the third party Service Provider domain.

The third party Service Provider is contacted and an agreement is made between the two Service Providers to perform service level tracing to identify the fault that causes the service failure.

5.8.2 Actors

End-user
- The end-users who use a mobile device to consume a service provided by their Service Provider.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

Service Provider
- Develops the service business logic which is realized as a complex set of interactions between components deployed by both the Service Provider and externally by a Third Party Service Provider;
- Supports the necessary network infrastructure to support and provide services to their customers. The Service Provider may also integrate services from a third party Service Provider to enhance their services portfolio;
- They are their end-user's first point of contact for any service related queries, e.g. subscription, payment and billing related, and service specific queries or issues;
- Manages and maintains the day-to-day operations of the Service Provider's network and supporting services;
- Designs, develops and implements the service logic that may be realized as a complex set of interactions between enablers deployed by both the Service Provider and externally by a third party Service Provider.

Third party Service Provider
- Supports the necessary network infrastructure to support and provide services to their customers, e.g. the Service Provider. The third party Service Provider has a contractual agreement with the Service Provider to develop and host services on behalf of the Service Provider;
- Manages and maintains the day-to-day operations of the third party Service Provider's network and supporting services;
- Designs, develops and implements the service logic that may be realized as a complex set of interactions between enablers deployed by both the third party Service Provider and by a Service Provider;
- Liaise between the Service Provider (their customer) and the third party Service Provider.

5.8.2.1 Actor Specific Issues

End-user

- Consumes the services they have paid for and expects to use the services with consistent and expected level of quality.

Service Provider

- Ensure developed service meets acceptance-test quality; they take it through acceptance test in the shortest time possible so that the service is then available to the end-users;
- Offers Services to their end-users and guarantees a quality of service for the services that are consumed;
- The primary contact point for the end-user. They need to ensure that the customer's complaints are dealt with in an effective and efficient manner to maintain customer satisfaction;
- Ensure it has all means/tools to be able to manage the services and solve/address possible issues an end-user may experience or which may impact end-user experience;
- Ensures that their service designs meet the expected market needs in terms of time-to-market and quality of service, and end-user expected service quality, e.g. service works first time and is simple use etc;
- They need to able to make the necessary corrections to the service logic whenever a service fault is detected.

Third Party Service Provider

- Offers Services to their end-users (e.g. Service Provider) and guarantees a quality of service for the services that are consumed;
- Ensure it has all means/tools to be able to manage the service and solve/address possible issues an end-user may experience or which may impact end-user experience;
- Ensures that their service designs meet the expected market needs in terms of time-to-market and quality of service, and their customer's expected service quality, e.g. service works first time and is simple use etc;
- They need to able to make the necessary corrections to the service logic whenever a service fault is detected;
- Needs to ensure that the customer's complaints (e.g. those of the Service Provider) are dealt with in an effective and efficient manner to maintain customer satisfaction.

5.8.2.2 Actor Specific Benefits

End-user
- Consumes the services they have paid for and gets consistent and expected quality of service.

Service Provider

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

- Able to achieve a quicker debugging phase of a new version of a service;
- Able to maintain and improve the quality of service for the services that their end-users consume;
- Maintains and improves level of customer satisfaction through efficient and effective resolution of end-user complaints. It will help in providing more quality fault related information to the appropriate support teams within the Service Provider when a fault occurs;
- Confidence that issues can be tracked, isolated and corrected in an effective and efficient manner. This enables the process of fixing and debugging the issues more quickly. It will also help in providing more quality fault related information not only to other appropriate support teams within the Service Provider but to third party Service Providers.

Third Party Service Provider

- Able to maintain and improve the quality of service offered to their end-users, and other customers including other Service Providers;
- Confidence that issues can be tracked, isolated and corrected in an effective and efficient manner. This enables the process of fixing and debugging the issues more quickly. It will also help in providing more quality fault related information not only to the other appropriate support teams within Service Provider but to third party Service Providers.

5.8.3 Pre-conditions

- Third Party Service Provider has made the necessary business and technical arrangements with the Service Provider in order to be able to develop and host the application on behalf of the Service Provider.
- The Service Provider, in conjunction with the third party Service Provider, have developed the service logic that enables the Service Provider's end-users to consume the service, when the third party Service Provider hosts the application.
- End-user has an account with the Service Provider and their subscription is fully paid.
- The Service Provider have the right type of mobile device to be able to replicate the service failure.
- The end-user is able to replicate the service failure.
- Both the Service Provider and the third party Service Provider network infrastructure supports Service level tracing. This also means that when a device is marked by the Service Provider, any calls/sessions invoked (until when the Service Provider unmarks the device) from that device initiates service level tracing in both the Service Provider and the Third Party Service Provider domain.
- The Service Provider and the third party Service Provider have agreed to pass trace information from the Service Provider domain to the third party domain in order to resolve faults associated with applications hosted by the third party.

5.8.4 Post-conditions

- The Service Provider replicates the service failure.
- The Service Provider and the third party Service Provider retrieve the logged service trace information from their respective networks and together they identify the fault that causes the service failure.
- The cause of the service failure is identified as being part of the Service logic within the third party Service Provider's network. The fault that causes the service failure is rectified.
- The Service Provider's end-users continue to consume their services getting a consistent and expected user experience each time.

5.8.5 Normal Flow

1. A number of Service Provider end-users experience a fault with a service they have paid for, and after a number of failed attempts contact their Service Provider.
2. The Service Provider performs a number of basic checks (e.g. checks end-user has enough money/air time credit) but is unable to identify the cause of error.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

3. The Service Provider replicates the service fault with their engineering test mobile phones and confirms that a service failure exists and then checks their network for error alarms and examines the available statistical information associated with the key network nodes for signs of unusual nodal behaviour. However, they are still unable to identify the cause of failure.
4. Instead of contacting the end-users, the Service Provider marks their test mobile phone and then replicates the service failure.
5. Each node (e.g. component) in the Service Provider network logs all associated trace information.
6. The Service Provider retrieves all logged service level trace information from all nodes in their network. The Service Provider analyses the retrieved trace information and confirms that the cause of fault is neither related to a network transport failure nor a network node failure. However, they do identify that the cause of the service failure may reside in the service logic within the third party Service Provide domain, as they are the providers of the application.
7. The Service Provider escalates the fault to the third party Service Provider.
8. An agreement is made by both the Service Provider and third party Service Provider to initiate Service Level tracing using the Service Provider engineering test mobile phone.
9. Again, the Service Provider marks the test mobile phone and repeats the failure case.
10. All nodes (component) in the Service Provider and the third party Service Provider network logs all associated trace information.
11. The Service Provider and third party Service Provider each retrieve logged information from their respective networks.
12. The Service Provider and the third party Service Provider analyses their retrieved trace information for the cause of the fault.
13. The fault is identified as residing in the third party network and is fixed by the third party Service Provider.
14. After the fault is fixed and the Service Provider repeats the test and confirms that the service is now working correctly.
15. The Service Provider contacts the end-users to inform them that the service is now working correctly.

5.8.6 Alternative Flow

5.8.6.1 Alternative Flow 1: Debugging a new version of a service before service is rolled into the live environment

1. A new version of a service is submitted for acceptance tests on the pre-production environment. The new version of the service gets installed.
2. Service Provider explicitly turns on service level tracing functionality both on all mobiles used for testing and on all deployed components in the pre-production environment traversed by this service in its execution flow (i.e., all those enablers whose interfaces are used as part of any service flow). It is assumed that service level tracing functionality may not always be “on” at all times for all deployed components: it is therefore envisioned a way for the Service Provider to turn it on/off with the desired level of granularity as needed.
3. Service Provider informs Third Party Service Provider to turn on service level tracing for their deployed component.
4. Acceptance test cases are run from the appropriate mobile device. Service level tracing information is collected for the specific test case session and reported to the Service Provider. (Depending on the service being analysed) Trace reports may contain information collected both from service business logic and/or components deployed within the terminal, and from service business logic and/or components deployed within the Service Provider network.
5. The result of the acceptance test case is NOT as expected by the test book. Looking at the service level trace report can therefore perform issue identification. This report covers the sequence of calls (and their details) between the device, the service business logic and each enabler interface (and possible sub-sequent calls from the enabler to other enabler interfaces when applicable). The report covers only the Service Provider owned parts of the service flow, while it shows the third-party Service Provider parts as a black box (knows what goes in and what come out, but has no visibility of details). The issue is found in the call by the service business logic to an enabler function for a deployed component of the Service Provider.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

6. With the help of the service level tracing report, the Service Provider produces a fix for the service business logic and a new version is made available for acceptance tests.

7. Step 4 is repeated and results are as expected by the test book. Service Level Trace report for this test-case is stored for future use in comparing trace reports from future regression testing.

8. Step 4 is then repeated for all acceptance test cases in the test book, including those ones where exception paths are experienced by the end-user. Service Level Trace reports for all these test-cases are stored for future use in comparing trace reports from future regression testing.

9. Service Provider accepts this service. Service Level tracing functionality may now be turned off in the pre-production environment at discretion of the Service Provider.

10. Service Provider rolls the service into the production environment.

11. Service Provider explicitly turns on service level tracing functionality both on all mobiles used for testing and on all deployed components in the production environment traversed by this service in its execution flow (i.e.: all those enablers whose interfaces are used as part of any service flow). It is assumed that service level tracing functionality may not always be "on" at all times for all deployed components: it is therefore envisioned a way for the Service Provider to turn it on/off with the desired level of granularity as needed.

12. A meaningful subset of the acceptance tests is performed on the production environment. All tests pass. Service Level Trace report from the production environment for these test cases are stored for future use in comparing trace reports from future regression testing on the production environment. Service Level tracing functionality may now be turned off at discretion of the Service Provider. Service is now available for end-users use.

5.8.6.2 Alternative Flow 2: The Service Provider is not able to replicate the fault

1. As per normal flow 1 and 2.
2. The Service Provider is unable to replicate the service fault with their engineering test mobile phones but they still checks their network for error alarms and examine the available statistical information associated with the key network nodes for signs of unusual nodal behaviour. However, they are still unable to identify the cause of failure.
3. As per normal flow 7 and 8.
4. An agreement is made by both the Service Provider and third party Service Provider to initiate Service Level tracing by asking the Service Provider's end-user to replicate the service failure.
5. The Service Provider Operations team marks the end-user's mobile phone and the end-user repeats the service failure.
6. As per normal flow 11 to 15.
7. The Service Provider contacts the end-users to inform them that the cause of service failure has been fixed and asks whether they continue to experience the same problems.

5.8.6.3 Alternative Flow 3: Third party Service Provider does not support Service Level Tracing

1. As per normal flow 1 to 8.
2. The Service Provider would like to ask the third party Service Provider to help in jointly identifying the cause of service failure by performing service level tracing but are aware that the third party Service Provider do not support the service level tracing functionality.
3. A decision is made to escalate the fault to the third party Service Provider and at the same time to provide a service complaint report that includes a concise version of the logged trace information.
4. The third party Service Provider analyses the service complaint report including the concise logged trace information.
5. Using their own service diagnostic methods, which take considerably longer, the third party Service Provider identifies and rectifies the cause of service failure.
6. The third party Service Provider informs the Service Provider that they have rectified the fault.

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

7. As per normal flow 15 and 16.

5.8.6.4 Alternative Flow 4: Third party Service Provider cannot identify the fault but suggest other reasons why the Service fails

1. As per normal flow 1 to 13.
2. The fault is not identified immediately but the third party Service Provider identifies a possibly issue with the Service Provider's transport capabilities and makes a suggestion to the Service Provider as to what could be the cause of fault.
3. After further investigation by both parties it is agreed that there needs to be service logic corrections in both the Service Provider domain and the third party Service Provider domain.
4. As per normal flow 15 and 16

5.8.6.5 Alternative Flow 5: Third party Service Provider cannot identify the fault but it is identified that the Service logic does not meet the SLA technical criteria

1. As per normal flow 1 to 13.
2. The fault is not identified immediately but it is identified that a service logic response time message exceeds the maximum time as specified by the Service Level agreement technical criteria.
3. The service logic is modified in the third party Service Provider domain.
4. As per normal flow 15 and 16.

5.9 Open Issues

None

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

6. Requirements (Normative)

6.1 High-Level Functional Requirements for life cycle management

Label	Description	Enabler Release
[LCM-HL-1]	The OSPE MUST support the ability to perform automated deployment (including configuration, installation, activation, publishing) and removal/withdrawal of components, applications and services.	
[LCM-HL-2]	The OSPE MUST provide mechanisms to support the replacement of multi-vendor components.	
[LCM-HL-3]	The OSPE MUST have a mechanism that makes available to the Service Provider all information associated to the existence of Services, applications and components, and the relationship between components and applications.	
[LCM-HL-4]	The OSPE MUST provide a mechanism that enables either a service, application or component to make known their changes in state (e.g., following a life cycle management function such as install, upgrade, or remove).	
[LCM-HL-5]	The OSPE MUST provide a mechanism that can be applied by the Service Provider in such a manner as to ensure that all preconditions (e.g. ensure that that no subscription are still active) and post-conditions (e.g. deletion or refresh all catalogue entries) are satisfied prior to a change of state (e.g. installation or removal) of a component, application or service.	
[LCM-HL-6]	The OSPE MUST provide a mechanism that can be applied by the Service Provider in such a manner to ensure that, at least, the following preconditions are checked before shutting down/withdrawing a service: 1. Make sure that there is not any available subscription to the service to be withdrawn; 2. Make sure no other component is making use of the service/applications.	
[LCM-HL-7]	The OSPE MUST provide a mechanism that can be applied by the Service Provider in such a manner to ensure that, at least, the following post conditions or actions are realized after shutting down/withdrawing a service: 1. Delete related catalogue entries; 2. Notify backend systems about the change (O&M, billing, subscription provisioning systems, etc.); 3. Delete dependencies that the withdrawn service/application had on the service provider components; 4. Update Service Packages containing the related service; 5. Notify authentication and policy enforcement mechanisms about the change (so that requests to the withdrawn service are rejected, and no policies are to be applied any more to it).	
[LCM-HL-8]	The OSPE MUST provide a mechanism that can be applied by the Service Provider in such a manner to ensure that, at least, the following post conditions or actions are realized after deploying a new service: 1. Catalogue entries are updated; 2. Service Packages are updated; 3. Notify backend systems about the change (O&M, billing, subscription provisioning systems, etc.); 4. Update information about the dependencies that the new service/application will have on the service provider components; 5. Notify authentication and policy enforcement mechanisms about the	

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

	change (so that requests to the new deployed service may be accepted, and policies for the new service may be applied).	
[LCM-HL-9]	Each component in the OSPE MUST support the capability to allow the Service Provider to perform life cycle functions, e.g. install, upgrade, downgrade, stop and start, of the component.	
[LCM-HL-10]	The OSPE MUST have a mechanism that enables the components in the OSPE to make available their O&M information, for use by backend systems.	
[LCM-HL-11]	The OSPE MUST have a mechanism to support the discovery and registry of a component and an interface, application and Service.	
[LCM-HL-12]	The OSPE MUST provide the means to perform the configuration, registration, publication and activation (start and stop) of the components, applications and services within the OSPE.	

Table 1: High-Level Functional Requirements

6.1.1 Security

Label	Description	Enabler Release
[LCM-SEC-1]	The OSPE MUST support mechanisms that protects against security threats, e.g. the discovery and registration of unauthorized components, applications and services.	
[LCM-SEC-2]	Only authorized users MUST be allowed to manage the OSPE (e.g. system administrator).	

6.1.2 Charging

None identified

6.1.3 Administration and Configuration

Label	Description	Enabler Release
[LCM-AC-1]	The OSPE MUST support the authentication and authorization of components and applications making use of life cycle management interfaces.	
[LCM-AC-2]	The OSPE MUST provide a mechanism to manage data related with subscriptions to services (e.g. user profile information).	
[LCM-AC-3]	The Service Provider MUST be able to manage information related to the existence of components, applications and the relationship between components and applications.	
[LCM-AC-4]	Each component MUST expose through a standard interface the characteristics of that a component, which includes: 1. Supported Interfaces; 2. Version of the interfaces; 3. Configuration data.	
[LCM-AC-5]	If an error condition is experienced during a life cycle management process, e.g. the configuration or upgrade of a component, it MUST be possible for the Service Provider to reverse the attempted process ensuring that the state of that component is reversed back to the state before the process was initiated.	
[LCM-AC-6]	It MUST be possible to perform life cycle functions on execution instances of components and applications	

Table 2: High-Level Functional Requirements – Administration and Configuration Items

6.1.4 Usability

Label	Description	Enabler Release

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

[LCM-USE-1]	The OSPE MUST not prohibit different component upgrade models, e.g. live upgrade model, cluster upgrade model. (NOTE: To achieve this requirement a list of upgrade models will be identified and described within OMA).	

Table 3: High-Level Functional Requirements – Usability Items

6.1.5 Interoperability

None identified

6.1.6 Privacy

None identified

6.2 Overall System Requirements for life cycle management

Label	Description	Enabler Release
[LCM-OSR-1]	The Service Provider environment MUST support the ability to modify and replace the components and applications without disrupting ongoing sessions and transactions.	
[LCM-OSR-2]	In order to allow for continuous evolution of services, the Service Provider environment MUST provide a mechanism to allow concurrent interface versions, component versions and application versions to exist and be accessible in the Service Provider environment.	
[LCM-OSR-3]	If an error condition (e.g. failure of a service or component) is experienced during life cycle management, e.g. the installation, de-installation, upgrade, downgrade, configuration modification, the OSPE MUST provide information that allows the Service Provider to determine the cause of the error condition.	

Table 4: High-Level System Requirements

6.3 Component and interface requirements for life cycle management

Label	Description	Enabler Release
[LCM-COM-1]	Each component and associated interfaces in the Service Provider environment SHOULD support versioning mechanisms that allow both forward and backward compatibility, and to allow the requesting component to determine the version of an interface that belongs to a receiving component.	
[LCM-COM-2]	Each component in the Service Provider environment MUST expose a standardized interface to support the following management functions: 1. Install; 2. Refresh (e.g. to initiate a update current component data to the component catalogue) 3. Remove; 4. Stop; 5. Start; 6. Upgrade; 7. Downgrade. 8. Activation & deactivation.	
[LCM-COM-3]	Each component in the Service Provider environment MUST expose a standardised interface to support the following management functions:	

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

	1. Fault management (e.g. logging and SNMP traps); 2. Performance management (e.g. measuring, usage monitoring). 3. Security (e.g. permission settings); 4. Configuration management 5. Service Subscription Provisioning (data that may include user preferences for a service, as support of the subscription provisioning process).	

6.4 High-Level Functional Requirements for service level tracing

Label	Description	Enabler Release
[SLT-HL-1]	Each component with in the OSPE MUST support Service Level Tracing (SLT).	
[SLT-HL-2]	A device/component supporting Service Level Tracing (SLT) MUST support marking.	
[SLT-HL-3]	When a service is initiated from marked component/device and when that service matches the service indicated in the marking request, the component/device MUST indicate in the related outgoing messages (E.g. SIP Invite) that SLT is required.	
[SLT-HL-4]	The encoding of the SLT logged trace information MUST be defined in a standard manner across all components.	
[SLT-HL-5]	There MUST be several levels (e.g. amount or granularity) of logged information captured by a component.	
[SLT-HL-6]	All actors (e.g. the components of the Service chain the end-user's identity), and associated characteristics (e.g. component version, supported execution environment, application version), MUST be identifiable within the logged trace information that is retrieved from all components in a service chain.	
[SLT-HL-7]	The Service Provider or other authorised actor MUST be able to correlate the service to be traced, as indicated in the marking request, with trace information retrieved from across components of a service chain.	
[SLT-HL-8]	The Service Provider MUST be able to identify and distinguish between all trace activities initiated by multiple devices at a single component.	
[SLT-HL-9]	Service Level Tracing MUST apply to both control and user plane to aid in identifying issues related to, e.g. timing misalignments between the user plane and control plane.	
[SLT-HL-10]	The specification of SLT SHOULD be done in such a way as to maximise the chance for the SLT trace token to be passed through a service chain, which includes SLT non-compliant components.	

Table 5: High-Level Functional Requirements

6.4.1 Security

Label	Description	Enabler Release
[SLT-SEC-1]	Security MUST be applied to SLT to protect against security threats such as unnecessary service processing and/or fraudulent use of SLT.	
[SLT-SEC-2]	Security MUST be applied to ensure that only authorised actors are able to mark an end-user's device.	
[SLT-SEC-3]	All logged trace information pertaining to a specific SLT instance MUST be retrievable only by authorised Service Providers or authorised actors.	

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

6.4.2 Charging

Label	Description	Enabler Release
[SLT-CRG-1]	All services that are subjected to SLT MUST be indicated as such in the charging information (e.g. in the Call Data Record (CDR)).	

Table 6: High-Level Functional Requirements – Charging Items

6.4.3 Administration and Configuration

Label	Description	Enabler Release
[SLT-AC-1]	OSPE MUST provide a mechanism for an authorised actor (e.g. a Service Provider) to activate or de-activate tracing on a component whilst ensuring that the propagation of the trace indication is not prohibited.	
[SLT-AC-2]	OSPE MUST provide a mechanism to allow a Service Provider or other authorized actor to initiate a marking request.	
[SLT-AC-3]	OSPE MUST provide a mechanism for a Service Provider or any other authorised actor to unmark a marked device/component.	
[SLT-AC-4]	OSPE MUST provide a mechanism for a Service Provider or any other authorised actor to request a permission from an end-user to “mark” a device.	
[SLT-AC-5]	OSPE MUST provide a mechanism for a Service Provider or any other authorised actor to mark a device with or without the end-user’s permission.	
[SLT-AC-6]	In the case where it is not possible to mark the end-user’s device, OSPE MUST provide a mechanism that allows any authorised actor to initiate SLT at any specific component within a service chain.	
[SLT-AC-7]	OSPE MUST provide a mechanism that allows the Service Provider or any other authorised actor to specify the criteria that causes the marking request to take affect, i.e. the service to be traced and the time at which an indication for requiring SLT is included in a signalling message.	

Table 7: High-Level Functional Requirements – Administration and Configuration Items

6.4.4 Usability

None identified

6.4.5 Interoperability

Label	Description	Enabler Release
[SLT-IOP-1]	An SLT trace indication MAY (e.g. depending on Service Level Agreements between Service Providers) be propagated outside the boundaries of an OSPE, e.g. an SLT trace indication may be passed from a Mobile Operator’s network to a 3rd Party Service Provider network.	

Table 8: High-Level Functional Requirements – Interoperability Items

6.4.6 Privacy

Label	Description	Enabler Release
[SLT-PRV-1]	For Privacy requirements see Privacy for Mobile Services requirements [Privacy].	

Table 9: High-Level Functional Requirements – Privacy Items

6.5 Overall System Requirements for service level tracing

Label	Description	Enabler Release

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

[SLT-OSR-1]	The maximum number of simultaneous SLT test routines SHOULD be configurable by the Service Provider. However, the maximum number MAY also be influenced by legislation.	

Table 10: High-Level System Requirements

6.6 Component and interface requirements for service level tracing

Label	Description	Enabler Release
[SLT-COM-1]	A component's captured trace information MUST contain information including but not limited to: 1. A mechanism to determine the sequence of components and the sequence of captured trace information from each component within a Service Chain. 2. A mechanism to identify a specific instance of SLT; 3. Component characteristics (e.g. enabler Id, supported protocol and protocol version, key enabler performance indicators such statistics etc); 4. Incoming and outgoing service attributes (e.g. IP Port address, hostname, destination address etc); 5. Activity derived from a message containing the trace indication, e.g. an activity such as end user-visible events.	
[SLT-COM-2]	A component MUST support the activation and deactivation of Service Level Tracing (SLT) as requested by a Service Provider or any other authorised actor.	
[SLT-COM-3]	A component MUST propagate the indication for SLT onwards to other components within the Service chain (even if the outbound protocols are different from the incoming protocols).	
[SLT-COM-4]	Upon the reception of an SLT trace trigger request, the component MAY provide, other than that requested by the SLT trace trigger request, a different level (i.e. finer detail) and type of logged information (e.g. information that may not be directly associated with the service being invoked).	
[SLT-NI-1]	Each component MUST expose a standardised interface that allows for the retrieval of all captured SLT trace information or the retrieval of captured trace information pertaining to a specific instance of SLT (e.g. captured trace information on a component may have resulted from several test cases but only captured information associated with one specific instance of SLT is required to be retrieved).	
[SLT-NI-2]	All trace information MUST be logged by a common logging function. A common logging function MAY reside either on a component or as a standalone logging function within OSPE.	

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20090101-I]

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-RD-OSPE-V1_0	12 Dec 2003	All	Initial release.
	21 Jan 2004	Section 4 & 5	Incorporated approved contributions from the Singapore meeting
	25 Feb 2004	Section 1, 4, 5 & Appendix B	Incorporated approved contributions from the Beverley Hills meeting
	15 Mar 2004	Section 5.3	Added OMA-REQ-2004-0017R01 approved during Feb 27 conference call
	2 Apr 2004	Section 5.4 & 5.5	Added OMA-REQ-2003-0869R02 and OMA-REQ-2003-0872R01 approved at the Kansas City meeting
	4 Jun 2004	Section 4.4	Added OMA-REQ-2004-0249R01previously approved.
	20 Sept 2004	Section 5.6 to 5.10	Added contributions approved on the September 17 conference call as per the minutes OMA-REQ-2004-0870.
	3 Dec 2004	Section 4,5 and 6 and App B	Added contributions approved in Barcelona and remove Appendix B (now covered in section 6).
	6 Dec 2004	Section 3	Added missing definitions from 2004-0892R04
	13 Dec 2004	All	Editorial changes and removal of obsolete use-case (SV Requirements) following agreements during 2004-12-13 OSPE CC. Also, corrected document name.
	14 Dec 2004	All	Editorial changes to add bulleted numbers to requirements
	20 Jan 2005	All	Alignment of use-cases with agreed definitions. Merger of use-case A and D (Deploying new services in an existing environment); Removal of use-cases sub-actors; alignment of actors across use-cases. Editorial changes including the insertion of the requirements into boxes. A global change of "Enabler implementation" to "component". Inclusion of OMA-REQ-2005-0003R03; 0030R01; 0029.
	24 Feb 2005	All	Included agreed contributions: OMA-REQ-2005-0049R02; 0048R04.Rename of requirements identifiers (SCM<-> LCM); corrected requirements numbering.
	08 Mar 2005	6.1, 6.3, 3.2	Included agreed contributions: OMA-REQ-2005-0058R02; 0120R01; 0060R01; 0143R01.
	16 Mar 2005	3.2, 6	Included agreed contributions: OMA-REQ-2005-0118R03; 131, 133, 135, 136, 137, 138.
	21 Mar 2005	6	Included agreed contributions: OMA-REQ-2005-032R01, 134R01, and 141R01.
	15 Apr 2005	3, 6	Included agreed contributions: OMA-REQ-2005-0142R01, 174R01. Also included previously agreed requirements as described in 0058R02.
	26 Apr 2005	3, 6	Included agreed contributions: OMA-REQ-2005-0231R01 and 0233R01. Also, a re-ordering of the SLT requirements was performed.
	27 Apr 2005	-	Correction to the document number.
	18 May 2005	all	Editorial corrections to RD based on comments received during formal review (see OMA-RDRR-OSPE-V1_0-20050513-D). Comments addressed include: 001, 002, 003, 008, 013, 015, 016 (URI), 020, 021. In addition, document names within the references have been corrected.
	23 May 2005	all	Updated to include input contributions: OMA-REQ-2005-283, 287R02, 288R01, 289, 291, 292R01, 0300, which were submitted in response to the submitted OSPE formal review comments.
Candidate Version OMA-RD-OSPE-V1_0	14 Jun 2005	n/a	Status changed to Candidate by TP OMA-TP-2005-0186-OSPE-RD-Package-For-Approval
Draft Version: OMA-RD-OSPE-V1_0	20 Apr 2009	All	Updated as per Consistency Review comments in OMA-CONRR-OSPE-V1_0-20090420-D Editorial fixes : styles, history box, references and definitions sorted
Candidate Version OMA-RD-OSPE-V1_0	19 May 2009	All	Status changed to Candidate by TP: OMA-TP-2009-0196-INP_OSPE_V1_0_RRP_for_Candidate_Approval

© 2009 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.