Architecture Requirements

Approved Version 1.0 – 21 Oct 2003

Open Mobile Alliance

OMA-RD_Architecture_V1_0-20031021-A

Contents

1. Scope (Informative)

This document describes the requirements on the OMA Service Environment, specified in the OMA Service Environment specification, v 1_0_4, 20030807 [OSE]. The requirements in this document have been gathered from a number of different sources, and amalgamated to a common format. They have not always been designed from use cases formally designed according to the template in this document.

2. References

2.1 Normative References

[OSE] "OMA Service Environment". Open Mobile Alliance
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

2.2 Informative References

[LIBERTY] "Liberty Architecture Glossary". Liberty Alliance Project.
URL:http://www.projectliberty.org/

[OMA-DICT] "Dictionary for OMA Specifications". Open Mobile Alliance
URL:http://www.openmobilealliance.org/

[RFC2828] "Internet Security Glossary". R.Shirey. May, 2000.
URL:http://www.ietf.org/rfc/rfc2828.txt

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

Address	For the purposes of this discussion, address refers to a URI
Authentication Assertion	See [LIBERTY].
Circle of Trust	See [LIBERTY].
Component	Hardware or software that is part of a functional unit
De-federate	See [LIBERTY].
Federate	See [LIBERTY].
Function	A specific purpose of an entity, or its characteristic action
Identity	See [LIBERTY].
Identity Provider	See [LIBERTY].
Principal	See [LIBERTY].
Pseudonym	See [LIBERTY].
Service Composability	The capability to assemble enablers or services in various combinations to produce new enablers or services.
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).
Single Log-Out (SLO)	See [LIBERTY].
Single Sign-On (SSO)	See [LIBERTY].
Trust	The extent to which someone who relies on a system can have confidence that the system meets its specifications, i.e., that the system does what it claims to do and does not perform unwanted functions. [source: RFC2828]

3.3 Abbreviations

O&M	Operations and Management
OMA	Open Mobile Alliance
OSE	OMA Service Environment
QoS	Quality of Service
SLO	Single Log-out
SSO	Single Sign-On
WAP	Wireless Applications Protocol

4. Introduction (Informative)

The OMA Service Environment (OSE) is foreseen to consist of a number of different components, outlined in the OMA Service Environment Specification [OSE]. It will also describe the interfaces to be used between those components. Service enablers developed according to OMA specifications will be required to conform to these specifications (e.g. use interfaces as defined in the specification).

This means that all service enablers defined by OMA (current and future) are in principle system elements of the architecture, according to the definition in the requirements template. Here, we are however constraining ourselves to the systems elements which will have to be defined as part of the OMA Service Environment specification, and these are discussed in section 6.3.

5. Use Cases (Informative)

Not Applicable.

6. Requirements (Normative)

6.1 High-Level Functional Requirements

We recognize some of the requirements in this section are testable and some are measurable. All of the requirements are verifiable.

1. The OMA Service Environment MUST enable deployment and use of OMA service enablers to allow for a wide variety of business models.
2. The OMA Service Environment MUST enable the use and deployment of any service enabler by any authorized actor.
3. The OMA Service Environment MUST facilitate the creation and deployment of services using OMA-defined service enablers.
4. The OMA Service Environment SHOULD enable the definition of components in such a way that functional overlaps between OMA enablers are minimized.
5. The OMA Service Environment MUST provide interfaces towards backend systems (e.g. charging, accounting, payment, provisioning, Operations & Management, etc.).
6. The OMA Service Environment SHOULD support the integration of service enablers, support systems and/or data sources that are not specified within the OMA.
7. The OMA Service Environment MUST support seamless user mobility, user equipment mobility and service mobility between multi-vendor and multi-domain environments irrespective of the underlying network infrastructure.
8. Using components developed according to the OMA Service Environment MUST NOT contradict or prevent any requirements imposed by legislation.
9. The Service Environment MUST provide for extensibility for future service enablers and compatibility between these service enablers.
10. The Service Environment MUST provide for the integration of existing service enablers defined by OMA with each other and with existing systems.
11. The OMA Service Environment MUST identify and define a set of functions that are common to most, if not all, use cases, and the ways these functions can be exposed and shared. Where such functions have been defined all OMA-specified enablers MUST use them.
12. The OMA Service Environment MUST be valid for any kind of service (e.g. messaging, WAP, location, “IN”-like services, corporate services, etc.
13. The OMA Service Environment MUST be suitable for services focused on any kind of users or segments, including pre-paid, post-paid, corporate users, mass market, etc.
14. The Service Environment SHOULD enable component reusability.
15. If authorized by a Principal, service enablers, services, service providers or other actors MUST be able to interact with other service enablers, services, or service providers on the behalf of the Principal. For example, the OMA Service Environment MUST support the mechanisms to allow a Principal to delegate consent to an Identity Provider, allowing that Identity Provider to authorize federation of that Principal’s identity at multiple Service Providers.
16. When authorized, Principals MUST be able to set policies (e.g. charging policies and privacy policies) on any request (including discovery).

17. The OMA Service Environment SHOULD support options for the choice of party for handling authentication, charging and/or storage of user profiles.

18. The OMA Service Environment MUST NOT assume network connections are permanent or long-lived.

6.1.1 Security

1. The OMA Service Environment MUST provide mechanisms for authentication of users, applications and third-party service providers, and authorization for the use of service enablers across and within service provider domains.

2. The OMA Service Environment MUST enable a Principal to authorize a service enabler or service provider to execute actions on its behalf.

3. The OMA Service Environment SHOULD NOT disallow different trust models for brokered authentication assertions or for single authentication assertions.

4. The OMA Service Environment MUST allow optimisations if a requestor and responder are in the same domain i.e. trust domain).

5. The OMA Service Environment MUST enable single sign-on and single log-out to span enablers in a single domain or across multiple Service Provider domains. One-time authentication or a SSO MUST remain valid throughout a continuous session

6. The OMA Service Environment MUST support setting various strengths of security policies and SHOULD support a way for service providers to define and communicate authorization policies for enablers.

7. The OMA Service Environment SHOULD support a way to negotiate security settings between service providers.

8. The OMA Service Environment SHOULD provide a set of security functions (including methods and data models), which are common to all enablers and can be re-used by existing enablers and in the design of new enablers.

9. The OMA Service Environment MUST provide secure and confidential access to services and associated exchanges within and across networks and domains e.g. through methods such as encryption, integrity protection, non-repudiation, authentication (both mutual and one-way) and authorization.

10. The OMA Service Environment MUST be able to control access to enablers, irrespective of the network technology and domain of origin of the party attempting to access the enabler.

11. The OMA Service Environment MUST support a mechanism to federate and de-federate identity information across Service Provider domains.

12. The OMA Service Environment MUST provide mechanisms that ensure protection against security threats.

13. The OMA Service Environment MUST allow a Service Provider to request authentication confirmation from an Identity Provider either on behalf of itself or other Service Providers.

14. The OMA Service Environment MUST provide an interface between the authorization function and the charging enabler.

6.1.2 Charging

1. The OMA Service Environment MUST NOT preclude any charging models between different actors.

2. The OMA Service Environment MUST provide an interface where Accounting and Charging information is to be gathered.

6.1.3 Administration and Configuration

1. The OMA Service Environment SHOULD provide for the simplification of the services and service enablers life-cycle management by avoiding manual processes, need of integration due to lack of standards, etc.

2. Subject to authorization by the Service Provider, the OMA Service Environment MUST enable entities (e.g. enterprises) other than the service provider to upload applications, manage the service life cycle and manage devices according to the OMA Device Management requirements.

3. The OMA Service Environment MUST enable the communication of service monitoring data (e.g. performance measurements) between actors.

4. The OMA Service Environment SHOULD enable easy administration and configuration of users and services.

5. The OMA Service Environment MUST provide the means to manage the activation, registration, authentication, and authorization of users and service components.

6. The execution or use of access and authorization functions SHOULD NOT impact the performance of services.

7. The OMA Service Environment SHOULD provide functions for the management of trust between the actors in the OMA environment.

8. The OMA Service Environment MUST provide a mechanism by which device and network information can be communicated to an authorized third-party (with respect to the information holder) in a manageable way. This mechanism MUST allow for the automated discovery of new devices and new characteristics in existing devices.

9. The OMA Service Environment MUST provide a mechanism to enable third-parties to obtain an identification for an end-user who uses a particular device to access authorized third-party applications.

10. The OMA Service Environment MUST provide a mechanism to allow third-parties to discover the device(s) currently used by an end-user, if registered on a network (e.g. where to send a notification to the employee).

11. The OMA Service Environment MUST provide a mechanism for an authorized third-party to discover the conditions for using a service enabler exposed by a particular service provider in a dynamic manner.

12. The OMA Service Environment MUST support a mechanism for service providers and other authorized actors to enforce the conditions for use of a service enabler.

13. The OMA Service Environment MUST have a single logical point that handles subscriber and subscription information.

6.1.4 Usability

1. The OMA Service Environment MUST provide the means to simplify end-user service access and use.

6.1.5 Interoperability

1. The OMA Service Environment MUST define the data flows and interfaces between applications and enablers, and between enablers. These are the interfaces where interoperability is required.

2. The OMA Service Environment MUST NOT mandate any specific deployments.

3. The OMA Service Environment MUST support simplified (e.g., plug-in) and automated integration for enablers with each other.

4. The OMA Service Environment MUST provide a common mechanism for Provisioning of services, service enablers and user parameters.

5. The OMA Service Environment SHOULD provide a mechanism to manage and use policies (e.g. access policies, charging polices, service level agreements, etc.).

6.1.6 Privacy

1. The OMA Service Environment MUST provide a means to manage and enforce end-user privacy.

2. The OMA Service Environment MUST support the use of pseudonyms for the communication of Principal's identities between Service Providers (to enable traceability without disclosing the Principal's identity).

6.2 Overall System Requirements

See previous sections.

6.3 System Elements

1. The Service Environment SHOULD NOT preclude the deployment of service enablers in high-availability, high-uptime, scalable environments (e.g. By requiring implementation in ways which disable the use of the functions of this environment).
2. The Service Environment MUST allow applications to make use of multiple enablers to create services (e.g. service composability).
3. The Service Environment SHOULD enable the definition of components in such a way that consistent design (e.g. reuse of data formats, reuse of components, etc) is encouraged.
4. The Service Environment MUST support the ability to simultaneously operate multiple versions (i.e. multiple instances, defined according to different releases of the OMA specifications) of an interface or API.
5. The Service Environment MUST provide a mechanism to control the QoS and the service quality of the behaviour of enablers.
6. The specification of a Service Enabler MUST be done in such a way that allows for scalable implementations.

6.3.1 General requirements on enabler interfaces

1. The interfaces to a Service Enabler MUST NOT constrain the functions of the enabler to a single domain.
2. When a Service Enabler is defined by OMA a standardized interface MUST be defined for the Service Enabler.

6.3.2 Common Directory / Registry

1. The OMA Service Environment MUST have a single logical access point (e.g. Common Directory) to handle: 1) registration, 2) discovery and 3) functions and data that handle information relevant to more than one single service enabler.

6.3.2.1 Interfaces to Common Directory / Registry

1. The OMA Service Environment MUST support Service Registration for Services visible to the end-user.
2. The OMA Service Environment MUST support Service Discovery for services visible to the end user.
3. The OMA Service Environment MUST support Discovery for an implementation of a Service Enabler.
4. The OMA Service Environment MUST support Registration for an implementations of a Service Enabler.
5. Within the OMA Service Environment it MUST be possible to register, discover, and retrieve information (e.g. a service enabler's address) using a resource identifier (e.g. a user identifier).

6.3.3 Network interfaces

1. The OMA Service Environment MUST define a common interface for the operations and management (O&M) of both common and service-specific enablers or applications (including service monitoring and end-to-end service delivery).

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-RD_Architecture_V1_0-20031021-A	21 Oct 2003	Initial document to address the basic starting point Ref TP Doc# OMA-TP-2003-0591-ARCH-RD

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
n/a	n/a	n/a	n/a