Parlay in OSE Architecture

Approved Version 1.0 – 31 Mar 2008

Open Mobile Alliance

OMA-AD-PIOSE-V1_0-20080331-A

Contents

Figures

1. Scope (Informative)

The PIOSE AD addresses how OMA OSE and OSA/Parlay architectures could be integrated and how components (in an implementation or realization) could coexist; it addresses integration of OSA/Parlay, Parlay X Web Services, and OSE.

The information contained in this AD is applicable to OMA working groups that are developing service enablers that may use other resources such as Parlay and Parlay X components. The AD focuses on how the OSE could take advantage of OSA/Parlay and Parlay X Web Services..

The specific OMA service enablers are not affected by this AD and are handled by the individual OMA working groups as usual. OSA/Parlay architecture [OSA] is not affected by this AD either.

2. References

2.1 Normative References

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

[OSE] OMA Service Environment, Open Mobile Alliance™, OMA-AD-Service_Environment-V1_0_4-20070201-A, http://www.openmobilealliance.org

[OMADICT] "Dictionary for OMA specifications", Open Mobile Alliance™, OMA-ORG-Dictionary-V2_6-20070614-A , URL:http://www.openmobilealliance.org

[PIOSE-RD] "Parlay/OSA in OSE Requirements", Open Mobile Alliance, OMA-RD-ParlayinOSE-V1_0-20070615-C, URL:http://www.openmobilealliance.org/

2.2 Informative References

[ParlayWP] White Paper, Parlay Group: "Comparing OMA OSE and Parlay Architectures" http://www.parlay.org/imwp/idms/popups/pop_download.asp?contentID=5741

[OSA] Open Service Access (OSA); Stage 2

E.g. 3GPP TS 23.198 (Release 6)
URL: http://www.3gpp.org/ftp/Specs/latest/Rel-6/23_series/

[ParlayX] Open Service Access (OSA);Parlay X Web Services; Part 1: Common
E.g. ETSI ES 202 391-1 (Parlay X2) and ETSI ES 202 504-1 (Parlay X3)

URL: http://www.parlay.org/en/specifications/index.asp

or 3GPP Equivalent:
Open Service Access (OSA);Parlay X Web Services; Part 1: Common

3GPP TS 29.199-1
URL:http://www.3gpp.org/ftp/Specs/latest/

Overview & Status of OSA API Specifications:

http://portal.etsi.org/docbox/TISPAN/Open/OSA/Overview.html

[Parlay] Open Service Access (OSA); Application Programming Interface; Part 1: Overview

E.g. ETSI ES 203 915-1 (Parlay 5) and ETSI ES 204 915-1 (Parlay 6)
URL: http://www.parlay.org/en/specifications/index.asp

or 3GPP Equivalent:
Open Service Access (OSA); Application Programming Interface; Part 1: Overview
3GPP TS 29.198-1

URL: http://www.3gpp.org/ftp/Specs/latest/

Overview & Status of OSA API Specifications:

http://portal.etsi.org/docbox/TISPAN/Open/OSA/Overview.html

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. This document is not intended to provide testable requirements to implementations.

3.2 Definitions

Application	Use definition from [OMADICT].
Enabler	Use definition from [OMADICT].
Interface	Use definition from [OMADICT].
OSA Application Server	An entity that provides the service logic execution environment for client applications using the OSA API as specified in 3GPP TS 29.198 [Parlay]. It’s an AS that utilizes and exposes for service creation purpose the OSA/Parlay and Parlay X interfaces.
OSA/Parlay GW	Synonym for Service Capability Server [Parlay] NOTE: From the viewpoint of applications, a Service Capability Server (SCS) can be seen as a gateway (GW) to resources. [Derived from: [Parlay]]
OSA/ Parlay Interface	Standardized Interface used by application to access service capability features. [Parlay]
Service	Use definition from [OMADICT].
Service Capability Feature (SCF)	Functionality offered by Parlay or Parlay X services
Service Capability Server (SCS)	Functional Entity providing OSA interfaces towards an application. [Derived from: [Parlay]]

3.3 Abbreviations

3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
API	Application Programming Interface
AS	Application Server
FW	Framework (OSA/Parlay)
GW	Gateway (OSA/Parlay)
OMA	Open Mobile Alliance
OSA	Open Service Access
OSE	OMA Service Environment
PIOSE	Parlay in OSE
SCF	Service Capability Feature
SCS	Service Capability Server

4. Introduction (Informative)

The Parlay in OSE architecture document, as its name implies, is intended to define the high level architecture that defines the coexistence and use of Parlay within the OSE. Its purpose is to describe how the OSE can take advantage of Parlay X APIs and Parlay APIs. The PIOSE provides clarification of the relationships between OMA enablers and OSA/Parlay and Parlay X Web Services API implementations within the OSE.

Several OMA enablers are exposing or using capabilities that can be supported through OSA/Parlay APIs or Parlay X Web Services API interfaces. The Parlay Group has together with 3GPP defined interfaces for functions that are similar to or overlap OMA enablers (e.g. presence, location). The PIOSE describes how OMA enabler implementations may take advantage of OSA/Parlay and Parlay X in particular to avoid overlaps. PIOSE defines how Parlay X Web Services interfaces and Parlay interfaces can coexist and be used with OMA enablers within the OSE.

The purpose of this is to help OMA working groups better understand how and where they can take advantage of these Parlay API and Parlay X API specifications to accelerate OMA Enabler work, when and where appropriate and in particular to help to avoid overlaps.

Below are some guidance given on how an OMA enabler implementation may take advantage of the Parlay X web services APIs and the OSA/Parlay APIs. Parlay X APIs and Parlay APIs implementations can be realized in OSE.

An OMA enabler can utilize Parlay in the following ways (non exhaustive list):

- Parlay X used as I0 WS binding to some appropriate enabler
- Parlay X may help define highly abstract technology neutral I0 interfaces
- Parlay API used as I0 binding to some appropriate enabler
- Parlay may help define abstract technology neutral I0 interfaces

4.1 Planned Phases

All the PIOSE requirements are planned to be fully met in this release. No future releases are currently planned.

4.2 Security Considerations

No additional security functionality, above and beyond OSE, is needed by this enabler.

5. Architectural Model

5.1 Dependencies

The PIOSE enabler is dependent on OSE [OSE]. No other dependencies are identified.

5.2 PIOSE Logical Architecture (Normative)

The logical architecture for the relationship between Parlay and the OSE specified in this document is illustrated in Figure 1.

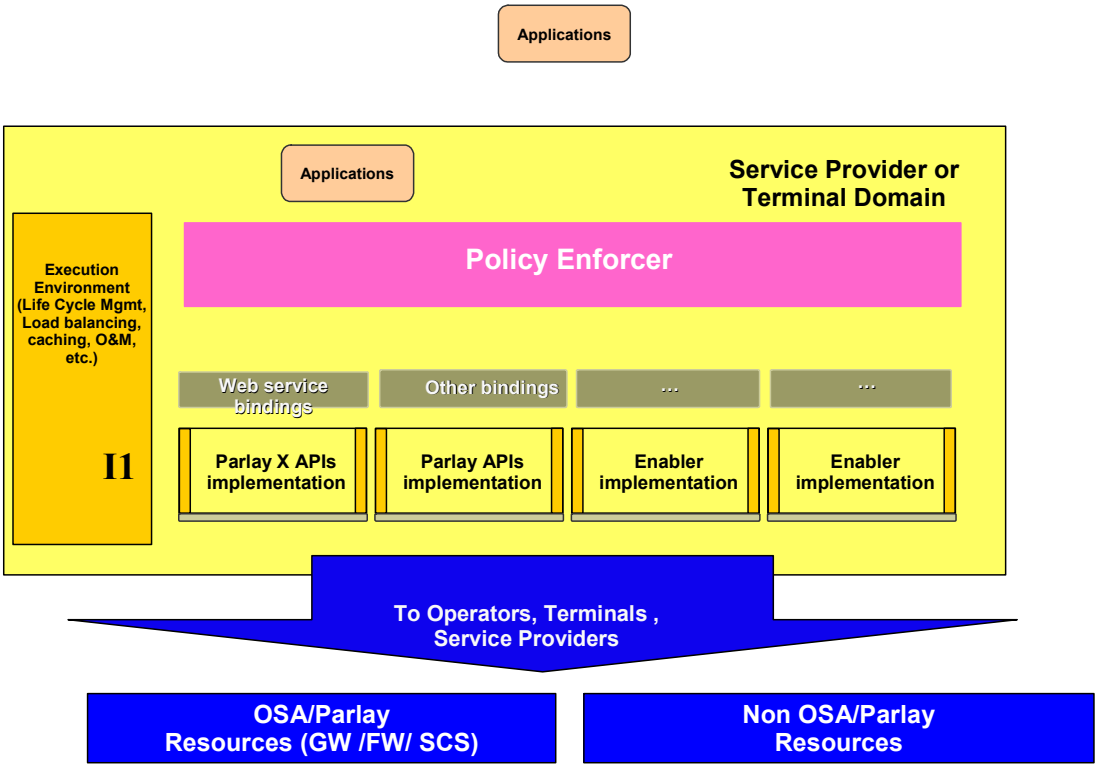

Figure 1: Logical architecture that illustrates the relationship between Parlay and OSE.

The OSE structure permits non-enabler implementations (such as a Parlay X or Parlay API implementation) at the same level as enabler implementations. Any underlying resource can be used to provide functions needed to support or realize those enabler or non-enabler implementations. For example, OSA/Parlay resources could be used for an OMA enabler implementation, or non-OSA/Parlay resources could be used by a Parlay X or Parlay API implementation

The elements in this architecture include:

- The OSE [OSE], associated to a service provider domain, and its elements:
  - Enabler implementations
  - Technology specific bindings
  - The Policy Enforcer function
  - The function of Execution Environment

- Resources including:
  - OSA/Parlay resources [Parlay]:
    - Parlay GW
    - Parlay Framework
    - OSA SCS
  - Non OSA/Parlay resources
- Parlay X APIs implementations
- Parlay APIs implementations

5.3 PIOSE Context Diagram (Normative)

The PIOSE context diagram in Figure 2 describes how Parlay can be used consistently with the OSE.

Figure 2 shows the conceptual OSE architecture including its three categories of interfaces, I0, I1 and I2 [OSE]. For the Parlay X APIs and Parlay APIs implementations the I0 interface represents a realization of Parlay X APIs or Parlay APIs. There exist multiple Parlay X API [ParlayX] and OSA/Parlay API [Parlay] specifications that expose different capabilities such as user location, call control, charging, presence, and terminal capabilities.

Furthermore, the PIOSE context diagram depicts the OSA/Parlay Resources and other resources (Non OSA/Parlay Resources) in the context of the OSE I2 interface category [OSE].

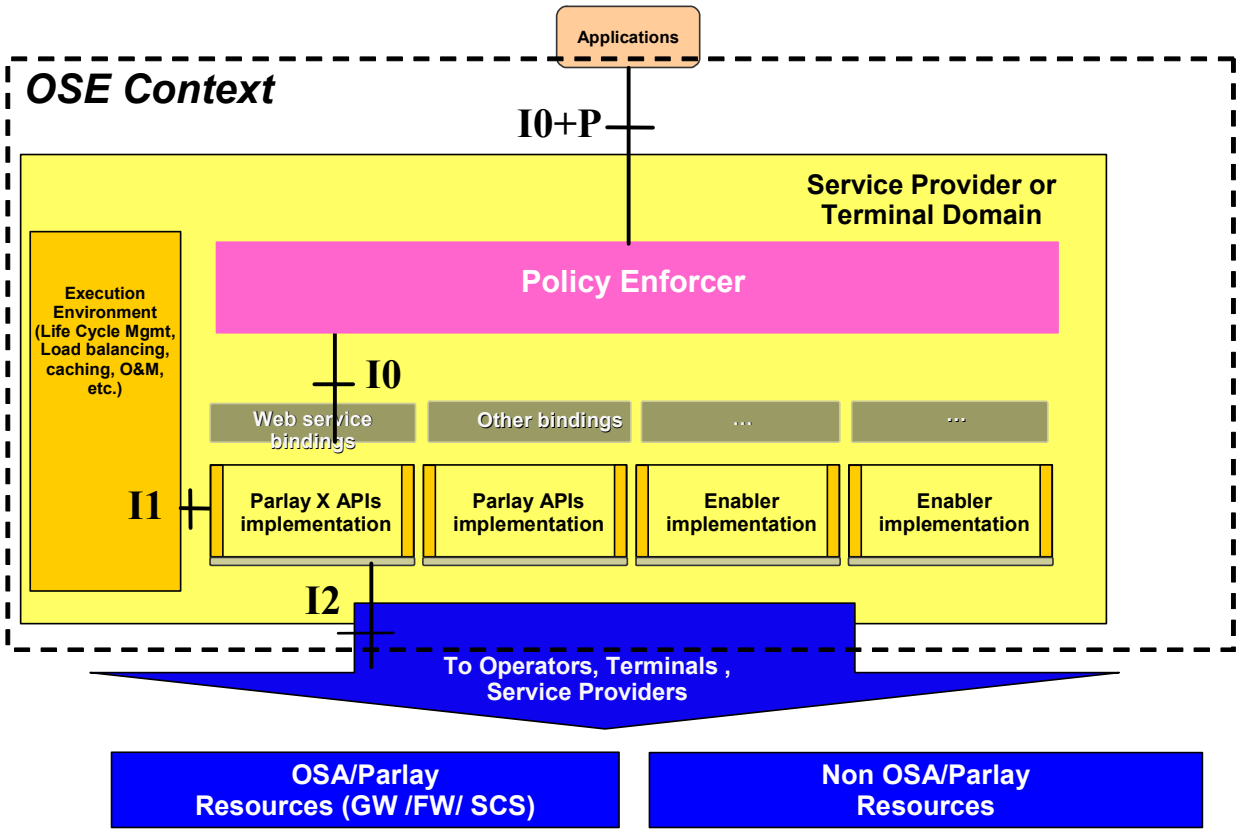

Figure 2: PIOSE Context Diagram

There are no constraints imposed by OSE on Enabler implementations regarding which resources and I2 interfaces that can be used. It is possible to provide the Parlay X and Parlay API implementations without using any underlying OSA/Parlay resources for that function.

A possible set of OSA/Parlay resource interfaces that correspond to I2 are the OSA/Parlay API specifications listed in [Parlay]. Notice that the specification of the Parlay and Parlay X APIs as well as OSA/Parlay Resources (e.g. SCS) is not within OMA's scope.

5.4 OSE and Parlay (Informative)

This section provides a clarification of the relationships between OSE, OMA enablers and OSA/Parlay and Parlay X Web Services specifications; this explanation should help OMA working groups better understand how and where they can take advantage of these specifications to accelerate OMA work, when and where appropriate.

5.4.1 The OMA Service Environment (OSE)

The OSE [OSE] is a conceptual and structured environment that includes OMA enablers, and interfaces to use them, plus underlying resources to support enabler implementations. Parlay and Parlay X capabilities is one such a group of resources described in this document.

5.4.2 The OSA/Parlay Service Environment

In collaboration with specification partners including 3GPP and ETSI, the Parlay Group has produced the Parlay APIs [Parlay], a set of technology independent functionally rich interfaces for a broad range of Telecommunications enablers including Call Control, Messaging, Location, Presence, Policy, Charging etc., and a set of abstracted Telecom Web Services, Parlay X [ParlayX].

The Open Service Access [**OSA**] defines an architecture that enables service application developers to make use of network functionality through open standardized interfaces, i.e. the OSA/Parlay APIs and Parlay X Web Services APIs. The network functionality is described as Service Capability Features (SCFs). The OSA/Parlay Framework (FW) is a general component in support of Service Capabilities and Applications. The concepts and the functional architecture for the OSA are contained in [**OSA**]. It may also be worth noting that implementation mapping guidance documents exist both for Parlay X APIs and Parlay OSA APIs for mapping to network protocols (~ I2 interface). An overview of the available Protocol mapping for Parlay X APIs and OSA APIs can be found in [ParlayX] and [Parlay] respectively.
Applications may use OMA enablers or may use OSA/Parlay resources directly or both.

5.4.3 Mapping of OSA/Parlay in OSE

In OSA [**OSA**] the OSA Application Server (AS) is connected via OSA API to OSA Service Capability Servers (SCSs) deployed in the network by the service provider. In the case of IMS, SCSs may connect to IMS through e.g. ISC interface.

The Figure 3 below clarifies how the PIOSE logical architecture can be seen as relating to different resources including the OSA/Parlay Resources and OSA architecture components described in OSA [**OSA**].The I0 interface can be exposed both to applications and enablers.

The Figure 3 clarifies how the OSE and OSE elements integrate and can take advantage of Parlay components and shows that the OSE can interact via the I2 interface category with OSA/Parlay resources (Gateways, framework and Service Capability Server (SCS)), and other resources.
There may be multiple ways to divide / organize physical deployment of OSA/Parlay resources and that the Figure 3 below is provided only to support explanation and may be used to cover logically different options but does NOT intend to imply or anticipate any particular Parlay X API or Parlay API Implementation in OSE.

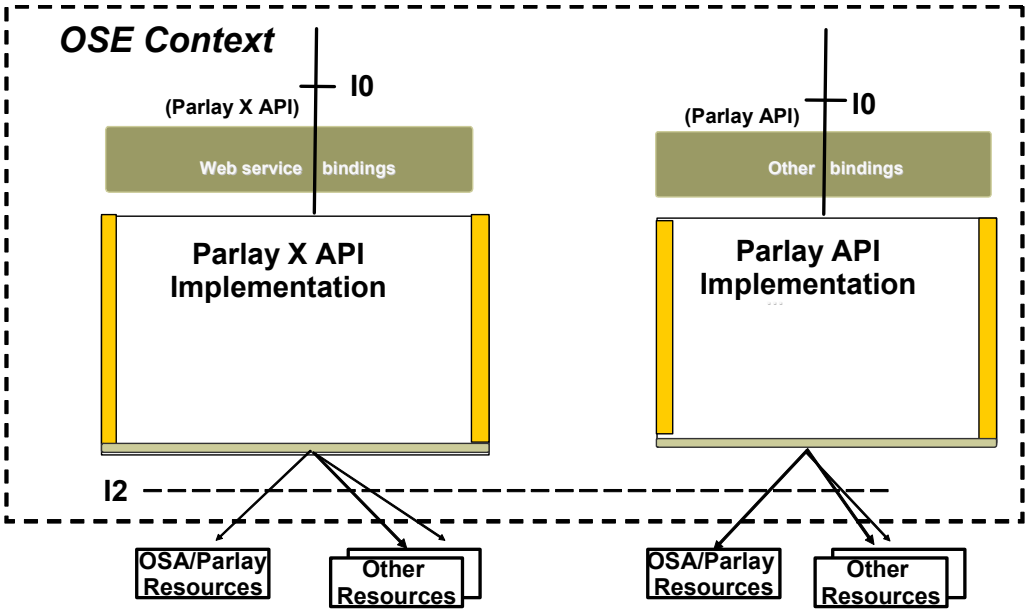

Figure 3: The Parlay X and Parlay API implementations in OSE and their logical relations to resources

5.4.4 OSA/Parlay Resources

The Figure 4 below depicts the logical functional view of OSA/Parlay resources, which can be deployed in different physical forms such as e.g. OSA/Parlay Gateways and SCSs. One of the OSA/Parlay SCSs is called the OSA/Parlay Framework, and is always present, one per network. This is further discussed in [**OSA**].

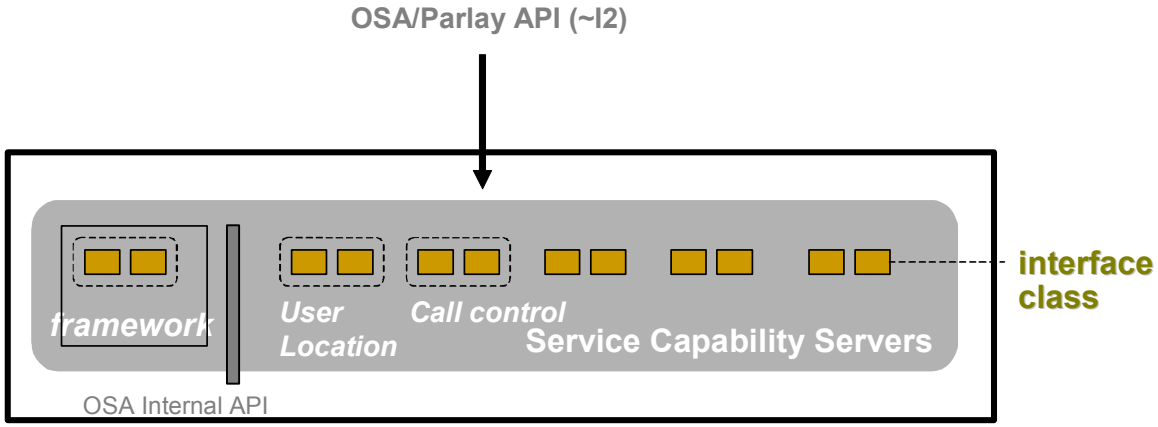

Figure 4: Logical view of OSA/Parlay Resources

The Figure 2 PIOSE context diagram shows that OMA enablers can expose Parlay interfaces (I0) without requiring to be implemented on OSA/Parlay resources. Any OMA enabler exposure of an interface I0 is independent of the underlying network. The OSE thus enables that the service provider would be able to provide the same applications on the different access networks.

Appendix A. Change History (Informative)

A.1 Approved Version 1.0 History

Reference	Date	Description
OMA-AD-PIOSE-V1_0	31 Mar 2008	Status changed to Approved by TP: OMA-TP-2008-0131-INP_PIOSE_V1_0_RRP_for_Final_Approval

Appendix B. PIOSE Deployment (Informative)

The PIOSE RD [PIOSE-RD] provides some high level requirements on integration of OSA/Parlay, Parlay X Web services and OSE. The purpose with the deployment examples below is to give some guidance on how the Parlay in OSE integration can be done and show the relationship between Parlay X APIs and Parlay APIs and OMA Enabler implementation. Several deployment scenarios are given below of possible Parlay in OSE integration and how OMA Enabler implementations can take advantage of the Parlay X and Parlay APIs implementations.

In all the figures, examples of elements in Network are SMS-C, MMS-C, WAP Push Proxy or an e-mail server, presence server, location server, CSCF server and charging server, etc. The dashed oval Network which includes the core control functions and Elements is out of the scope.

In all the figures: "~ I0" in the figures means that the corresponding Parlay X API and Parlay API are considered as equivalent to I0 in OSE.

B.1 Application Interact with Parlay Implementations

In this case, Parlay APIs and Parlay X APIs implementations can be considered at the same level as OMA enabler implementation in OSE. In such case, Parlay X API and Parlay API can be considered as playing an equivalent role to I0 in OSE: being exposed to the applications. The Parlay X APIs implementation uses solely web service as binding technology, whereas the Parlay APIs implementations may use other binding's technology. The Policy Enforcer (PE) provides the policy enforcement on exchanges with OMA Enabler implementation as well as Parlay X APIs and Parlay APIs implementations when needed.

In the Figure 5 below both OMA Enabler implementation and Parlay X APIs and Parlay APIs implementations and infrastructure are simultaneously involved, and PE applies on all exchanges to and from enablers and Parlay (X) API implementations.

Note: Parlay Policies may also apply to allow usage of the Parlay resources (SCS). But that is outside the scope of the OSE model. The service provider however may want to make sure that the policies are consistent at the two levels.

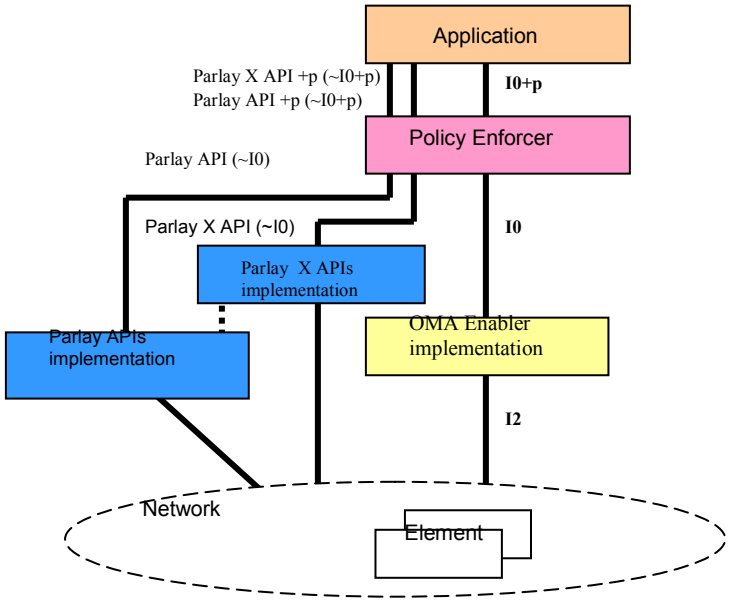

Figure 5: Integration of OMA Enabler and Parlay implementations including Policy Enforcement

B.2 Application interacts with OMA Enabler through Parlay Interface

This scenario addresses the case where an Enabler implementation is used to realize the function of Parlay/Parlay X API interfaces. Parlay/Parlay X exposes standard API to Applications, and this is realized through OMA Enabler implementation. An application using Parlay interface (Parlay X APIs and Parlay APIs) is deployed following the OSE. In that way, Parlay interfaces can be realized without Parlay infrastructure. Thus an existing Parlay application can be deployed also on other networks, where the service provider has deployed OSE but not Parlay infrastructure. The approach illustrated in the Figure 6 is provided as an example and not the only way to do that.

The example below shows a potential OMA Enabler implementation, which would use the OSA/Parlay API as I0 binding. A hypothetical use case would be a Parlay location service application, which would like to use an OMA Location Enabler implementation to realize the location service. In this scenario the OMA Location Enabler implementation would use Parlay User Location API as I0 binding. This would allow a Parlay X APIs to be exposed to applications using that same OMA enabler implementation, as illustrated in the figure below.

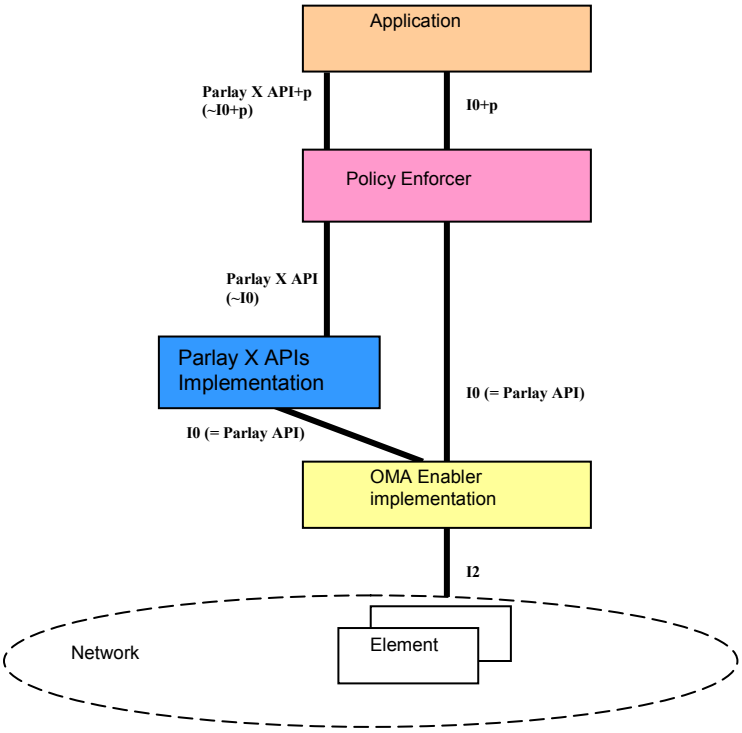

Figure 6: Parlay APIs as I0 by OMA Enabler implementation

B.3 Application Interact with OMA Enabler that utilizes Parlay Interface

In this case, Parlay X/Parlay is used to support OMA enabler implementation. The API which is exposed by Parlay is considered as I2 for OMA Enabler implementation. Thus the use of Parlay here is at the I2 level and hence outside the scope of OSE. Some new OMA Enabler implementations could use Parlay APIs or Parlay X APIs as I2, where there is no corresponding OMA enabler so far; for instance third party call control and conferencing.

The example below shows I2 for a Parlay APIs implementation, but a Parlay X APIs implementation could also be an option.

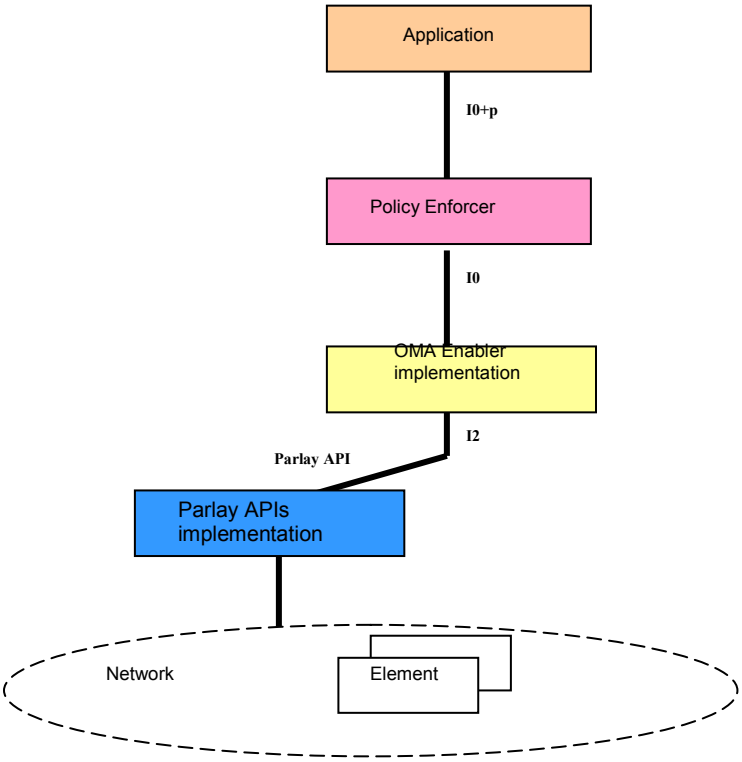

Figure 7: Utilization of Parlay APIs as I2 by OMA Enabler implementation

B.4 Application Interact with OMA Enabler or Parlay Implementation

When similar functions are supported by Parlay/Parlay X components and OSE enabler implementations at the same time, a mechanism (e.g. an API adaptor) that transforms the format from I0 of OMA enabler to the format of Parlay X API (or Parlay API) may become useful (Figure 8). A variant is when this mechanism transforms the format from Parlay API (or Parlay X API) of Parlay implementation to the format of I0 of OMA enabler (Figure 9).In the example in Figure 8 below the policies in the Policy Enforcer include performing API Adaptation. The API adaptor performs the necessary mapping transformation from the format of I0 of the OMA enabler to the format of Parlay X API.

Thus for example, an existing application which uses Parlay capabilities can use OMA enabler implementations as well. Potential use cases would be where OMA Enabler implementation and Parlay X / Parlay APIs implementations may perform the same logical functions. Possible usage of this mechanism could be in areas of e.g. Device Management, Presence, Charging, and Messaging, where a functional overlap exists that could make this feasible.

The first example in Figure 8 below shows a Parlay X APIs implementation, but applies equally well to a Parlay APIs implementation.

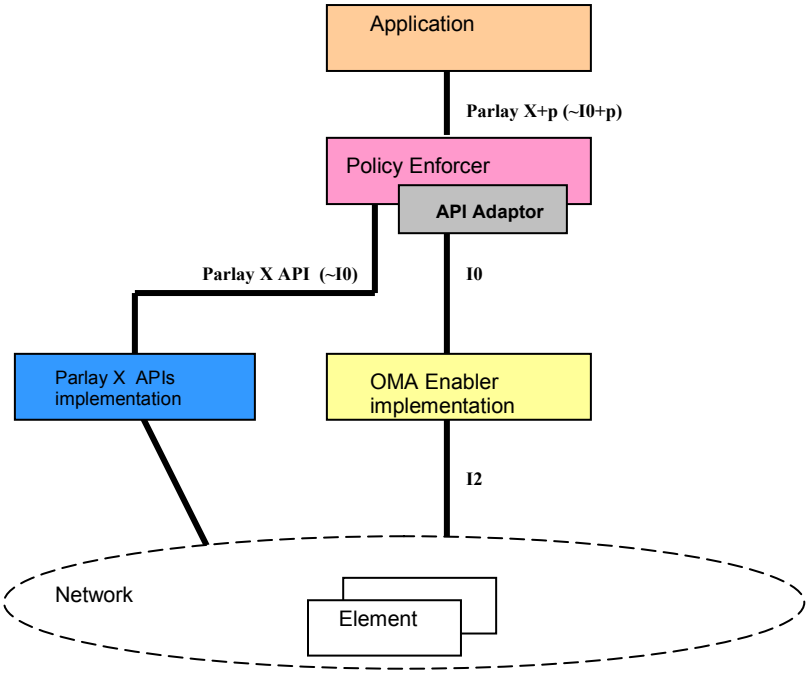

Figure 8: Application invokes OMA Enabler or Parlay X APIs implementation by Parlay X API

The second example in Figure 9 below shows another variation where the API adaptor transforms the format from Parlay API to the format of I0 of OMA enabler. The example shows a Parlay APIs implementation, but applies equally well to a Parlay X APIs implementation.

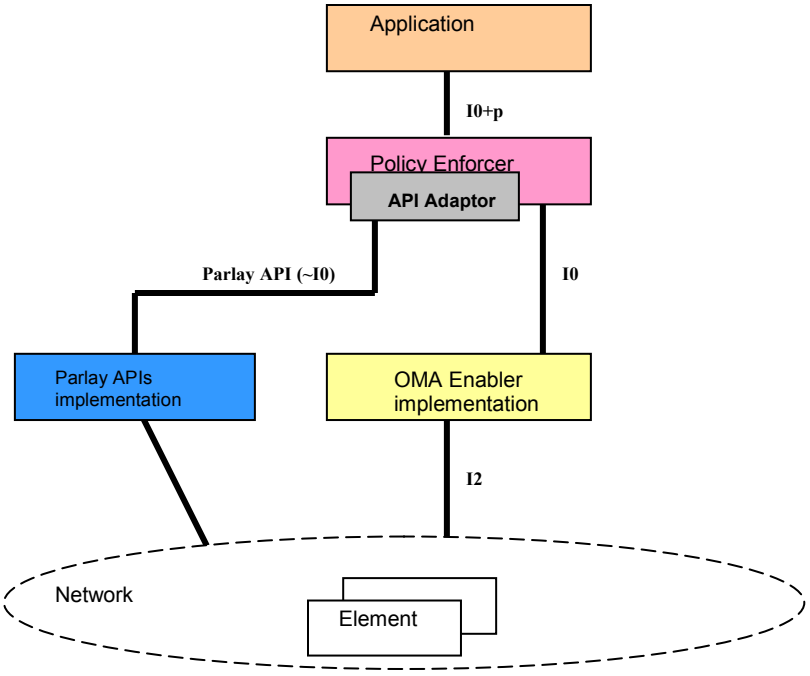

Figure 9: Application invokes OMA Enabler implementation or Parlay APIs implementation by I0

B.5 Dispatcher selects OMA Enabler or Parlay Implementation

When similar functions are supported by Parlay/Parlay X APIs implementations and OSE Enabler implementation at the same time, a dispatcher mechanism in the network is possible, which is depicted as "Service Dispatcher" in the figure below. The Service Dispatcher selects the corresponding Parlay X APIs or Parlay APIs implementation or OMA enabler implementation based on the request from the network and defined policy (e.g., workload balance). Furthermore, the request content may be modified if the selected implementation instance doesn't support the request such as to enable e.g., filtering in OMA presence which is not supported by Parlay PAM (presence and availability management). However, the Service Dispatcher specification is outside the scope of OSE model and thus depicted in Figure 10 as a network element.

The example below shows a Parlay X APIs implementation, but applies equally well to a Parlay APIs implementation.

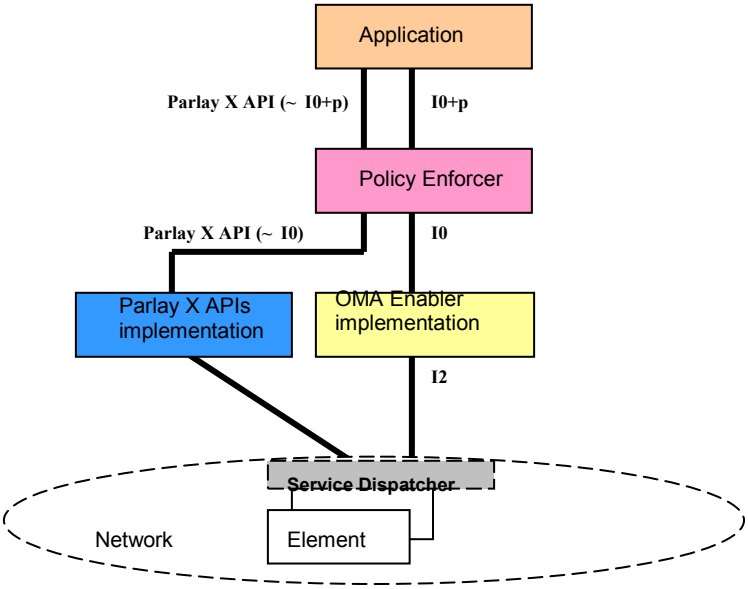

Figure 10: Dispatcher invokes OMA Enabler implementation or Parlay X APIs implementation