ETSI EG 201 988-4 V1.1.1 (2008-02)

ETSI Guide

Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Service Provider Access; Open Service Access for API requirements; Part 4: Version 4

Reference

DEG/TISPAN-01058-OSA

Keywords

API, architecture, interface, UML

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This ETSI Guide (EG) has been produced by ETSI Technical Committee Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN).

The present document is part 4 of a multi-part deliverable covering Service Provider Access; Open Service Access for API requirements, as identified below:

Part 1: "Version 1";

Part 2: "Version 2";

Part 3: "Version 3".

Part 4: "Version 4".

Introduction

The present document contains the Requirements capture for ETSI 4.0 "Third Party API" protocol specification: ES 203 915 series [1] and ES 202 391 series [2].

1 Scope

The present document contains the functional requirements for Open Service Access Requirements Version 4.0. The present document has been compiled in conjunction with Parlay and represents the sixth phase of the Parlay API. The ETSI and Parlay API have been specified and designed using the requirements identified both in the previously published Parts of this specification, Parts 1 through 3 and in the present document, Part 4. The requirements are intended to provide the necessary functionality for benchmark applications.

It is the intention that the new requirements should build upon the ETSI Phase 3.0 API and that of the Parlay 5.0 specification requirements, as described in EG 201 988-3 [3], and should be fully backward compatible. This means that any network operator implementing ETSI Phase 4.0 or Parlay 6.0 should be able to interwork with a client application provider implementing ETSI Phase 3.0 or Parlay 5.0. In other words ETSI Phase 4.0 and Parlay 6.0 will retain ETSI Phase3.0 and Parlay 5.0 as a complete subset. A full description of backward compatibility considerations is presented in clause 9 of ES 203 915-1 [4]. For any requirement that would result in an extension of, or would build upon, a part of the API specification set that is published jointly by 3GPP as well, in addition to ETSI and Parlay, the contributing companies are encouraged to submit their requirements to the 3GPP SA1 requirements process [5].

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[1] ETSI ES 203 915 (series): "Open Service Access (OSA); Application Programming Interface (API)".

[2] ETSI ES 202 391 (series): "Open Service Access (OSA); Parlay X Web Services".

[3] ETSI EG 201 988-3: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Service Provider Access; Open Service Access for API Requirements; Part 3: Version 3".

[4] ETSI ES 203 915-1: "Open Service Access (OSA); Application Programming Interface (API); Part 1 Overview (Parlay 5)".

2.2 Informative references

[5] ETSI TS 122 127: "Universal Mobile Telecommunications System (UMTS); Service Requirement for the Open Services Access (OSA); Stage 1 (3GPP TS 22.127)".

[6] ETSI TS 123 041: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Technical realization of Cell Broadcast Service (CBS) (3GPP TS 23.041)".

[7] ETSI TS 123 032: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Universal Geographical Area Description (GAD) (3GPP TS 23.032)".

3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ADQ	Application Driven Quality of Service
API	Application Program Interface
ASP	Application Service Provider
BM-SC	Broadcast Multicast-Service Centre
CBC	Cell Broadcast Centre
GGSN	Gateway GPRS Support Node
IP	Internet Protocol
IPTV	Internetworking Protocol Television
IVR	Interactive Voice Response
MMS	Multimedia Messaging Service
OSA	Open Service Access
SGSN	Serving GPRS Support Node
SMS	Short Messaging Service
TPC	Third Party Control

4 ETSI Phase 4.0/Parlay 6.0 API Domains

The Parlay/OSA API is an open, technology-independent, and extensible interface into networking technologies. The Parlay API is therefore applicable to a number of business and application domains, not just telecommunications network operators.

Examples of business domains that may use the API include:

- Third Party NGN Service Providers.
- Interactive Multimedia Service Providers.
- Corporate Businesses.
- Small Businesses.
- Residential Customers.
- Network Operators.

All of these businesses have networking requirements, ranging from simple telephony and call routing to call centre's, virtual private networks and fully interactive multimedia.

4.1 Requirements on interfaces at different levels of abstractions

As originally defined in clause 6.5 of EG 201 988-3 [3], the OSA-defined functions may be accessed through interfaces at different levels of abstractions and according to different programming formalisms, in addition to those defined in the previous Releases. Accordingly, ETSI Phase 3.0 and Parlay 5.0 is realized in two specifications sets:

- OSA APIs (Parlay 5).
- OSA Parlay X 2 Web Services.

For ETSI Phase 4.0 and Parlay 6.0, the requirements described in the present document will likewise be realized in two specifications sets:

- OSA APIs (Parlay 6): i.e. ES 203 915 series [1].
- OSA Parlay X Web Services: i.e. ES 202 391 series [2].

Guidelines have been adopted to determine which of the two abstraction layers provides the appropriate domain to realize the requirements described in the present document.

4.2 Parlay X Web Service Guidelines

The interfaces represented by the Parlay X Web Services should be **powerful yet simple and highly abstracted.** The following rules serve as guidelines for realizing requirements using Parlay X Web Services. Exceptions to these rules will be considered if they are justified by simplicity or completeness of the API, and if the resulting specification is sufficiently differentiated from related specifications.

- A Parlay X Web Service specification will be a functional abstractions of a Parlay/OSA specification. Where a functionally overlapping Parlay/OSA specification exists, the Parlay X Web Service specification will be an abstraction of the Parlay/OSA specification.
- Parlay X Web Service specifications should offer a coarser granularity level (e.g. measured as the relative size, level of detail, or depth of penetration), and contain less than half the methods of the equivalent Parlay/OSA specifications.
- Parlay X Web Service specifications should not mandate the maintenance of state.
- Parlay X Web Service specifications should not contain asynchronous message exchange, but often include event notification.
- Parlay X Web Service specifications should never imply detailed protocol knowledge.
- Parlay X Web Service specifications should be functionally self-contained from the developers point of view.
- For any Parlay X Web Service specification, 80 % of the above rules should be met.

5 Proposed enhancements to existing Interfaces

5.1 General requirements

5.1.1 Backwards Compatibility/Deprecation - Parlay/OSA APIs

A full description of backward compatibility considerations is presented in clause 9 of ES 203 915-1 [4].

5.1.2 Backwards Compatibility/Deprecation - Parlay X Web Services

For OSA Parlay X 3 Web Services it is desirable, **but it is not considered necessary**, to retain backwards compatibility with the existing OSA Parlay X 2 Web Services specifications. This is because the existing specifications are immature and there are limited implementations and deployments to date. This provide an opportunity to correct identified shortcomings, and in so doing provide a solution approach that will enable a richer set of applications whilst being application agnostic in nature.

5.2 Call Session Control

Issues and Motivation:

Operators and vendors desire to extend the call control capabilities of the Parlay X Web Services. The suggested approach to the evolution of Parlay X Call Control builds on the agreed view of keeping the Parlay X web services true to the design goal of "Separation of Concerns" and to avoid bundling of functionality where possible. This approach is **not** backward compatible with the existing Parlay X Call control-related web services, but this is an acceptable trade-off as noted in clause 5.1.2.

Requirements Description:

- Call Participant Control. The ability to add, remove, transfer call participants, for either application or network initiated calls.
- IVR Interaction. Application should be able to request IVR Interaction on a call. It should include simple Play Announcement and Play and Collect information capabilities for both network and application initiated calls.
- Additional Call Notification events. The following are specifically identified:
    - Call Progress.
    - Call Setup failure.
    - Call Party Disconnect.
    - Call Party Answer.
    - Call Rejected.
    - Media Changed.
- Deassign Call Control. The ability to stop an application from receiving notifications on a specific call.
- Media Control. The ability to manipulate the media on either an application or network initiated call.

Proposed Solution and Further Considerations:

These functions are currently implemented in the Parlay/OSA APIs.

These functions are also viewed as appropriate for implementation at an abstraction level consistent with that of the Parlay X Web Services.

Figure 1 shows the conceptual principles of how the involved call control elements relate.

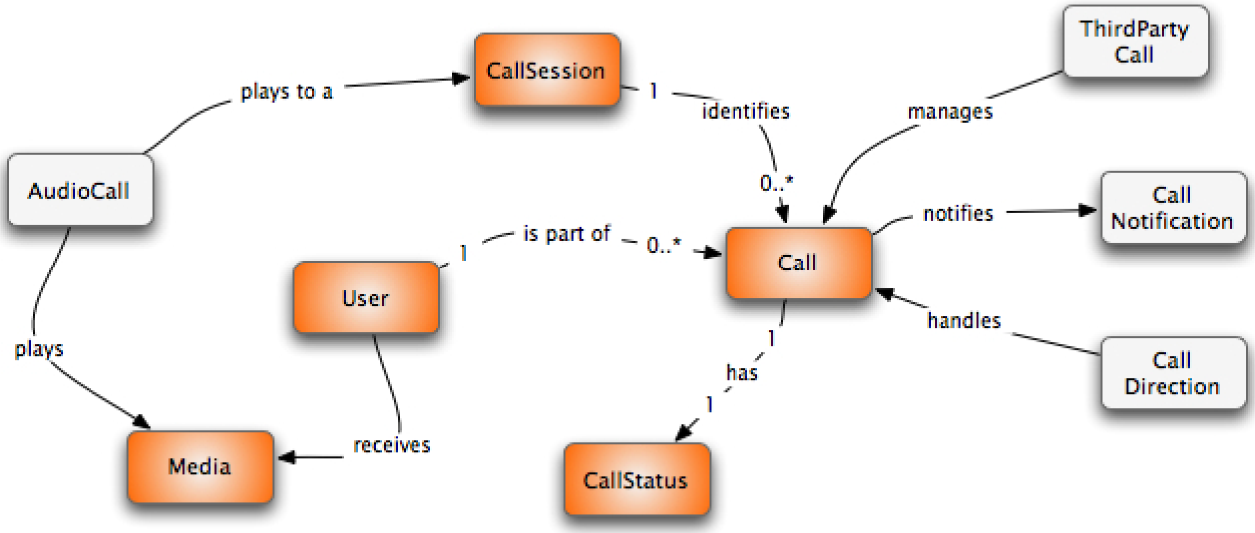


Figure 1

CallSession: The CallSession object can be used to interact with ongoing calls. It will also serve as a placeholder for a set of calls. It can be viewed as a call "context" (uniquely identified) to/from which participants can be added/removed.

ThirdPartyCall: The Third Party Call Web Service will now be the only place where application initiated call processing can and should be performed. This implies removing the call initiation from, for instance, the AudioCall web service. **CallNotification** and **CallDirection** interfaces are updated to support the CallSession concept.

The ThirdPartyCall MakeCall operation will further be enhanced with an optional parameter to indicate which CallSession the call is to be assigned to. If this parameter is not included in the request, the MakeCall will operate as in previous versions of Parlay X.

ThirdPartyCall provides the ability to setup a call session, add and delete a call participant, transfer a call participant from one call session into another call session, determine the status of an individual call participant or a complete call session, and finally to end a call session.

AudioCall: For the AudioCall service, it is proposed change the name to MediaCall and add a PlayVideoMessage operation to complete the set of operations. The service will now also change its behaviour such that it can only act on already established CallSession objects. This keeps a clear distinction between where the call control lives and where the media capabilities are added.

AudioCall will also be extended so that it can manipulate media on an ongoing call (either network or application initiated) and IVR interactions can occur.

User: A user can be part of many calls, it is important to note that the concept of a user may involve resources such as voice xml processing equipment and text-to-speech engines.

5.3 Scheduled Short and Multimedia Message Transmission (#6P11)

Issues and Motivation:

Operators desire the ability to send a short or multimedia message to a large set of subscribers. In addition, operators would like to provide third parties with the ability to create and schedule transmissions of large sets of message to increase messaging infrastructure usage.

The current messaging interfaces provide immediate message transmission only and do not allow applications to schedule message transmission during periods when network messaging resources are less utilized.

The commercial motivation is the ability to increase the volume of successfully delivered messages and the end-user application experience.

Scenarios:

Not applicable.

Requirements Description:

- Schedule the transmission of short messages to multiple destinations.
- Schedule the transmission of multimedia messages to multiple destinations.
- Retrieve the status of a scheduled message transmission request, including the number of messages successfully sent if a scheduled request is in progress or has completed.
- Cancel a scheduled message transmission request:
  - An ability to cancel a scheduled message transmission request has greater business value than an equivalent function for the existing immediate message transmission function. This is because this new function will typically be used to request transmission of large sets of messages well in advance of the scheduled time, resulting in a higher probability of a successful cancellation result and a reduction in unnecessary messaging resource usage.

Proposed Solution and Further Considerations:

This function is viewed as appropriate for implementation at an abstraction level consistent with that of the Parlay X Web Services. This function can also be mapped to the existing Parlay/OSA APIs.

5.4 Charging

Issues and Motivation:

The ability to split a charge between multiple end user accounts.

Scenarios:

A multi-player gaming application, where all participants share in the cost.

Requirements Description:

- Support split charging.

Proposed Solution and Further Considerations:

This function is currently implemented in the Parlay/OSA APIs.

This function is also viewed as appropriate for implementation at an abstraction level consistent with that of the Parlay X Web Services.

5.5 Account Management

Issues and Motivation:

Desire to take proactive measures (e.g. recharging) when an account balance falls below a threshold. Also the ability to monitor end user account activity: charging and recharging.

Scenarios:

Not applicable

Requirements Description:

- Support account balance related event notifications: charge, recharge, low balance.

Proposed Solution and Further Considerations:

This function is currently implemented in the Parlay/OSA APIs.

This function is also viewed as appropriate for implementation at an abstraction level consistent with that of the Parlay X Web Services.

5.6 Presence

Issues and Motivation:

Currently the Parlay X Presence Web Service provides information to the application on the various means of communicating with a presentity: e.g. voice, chat, SMS, video etc. However the status of each such means of communication is not available to the application which limits development of richer applications.

Scenarios:

Not applicable.

Requirements Description:

- For each of the possible means of communicating with a presentity, provide the application also with the current status of that communication means.

Proposed Solution and Further Considerations:

This function is viewed as appropriate for implementation at an abstraction level consistent with that of the Parlay X Web Services. This function can also be mapped to the existing Parlay/OSA APIs and SIP/IMS networks.

The proposed solution would add a status element (on, off, busy) to the CommunicationMeans structure.

6 New interfaces and areas of involvement

6.1 Message Broadcast

Issues and Motivation:

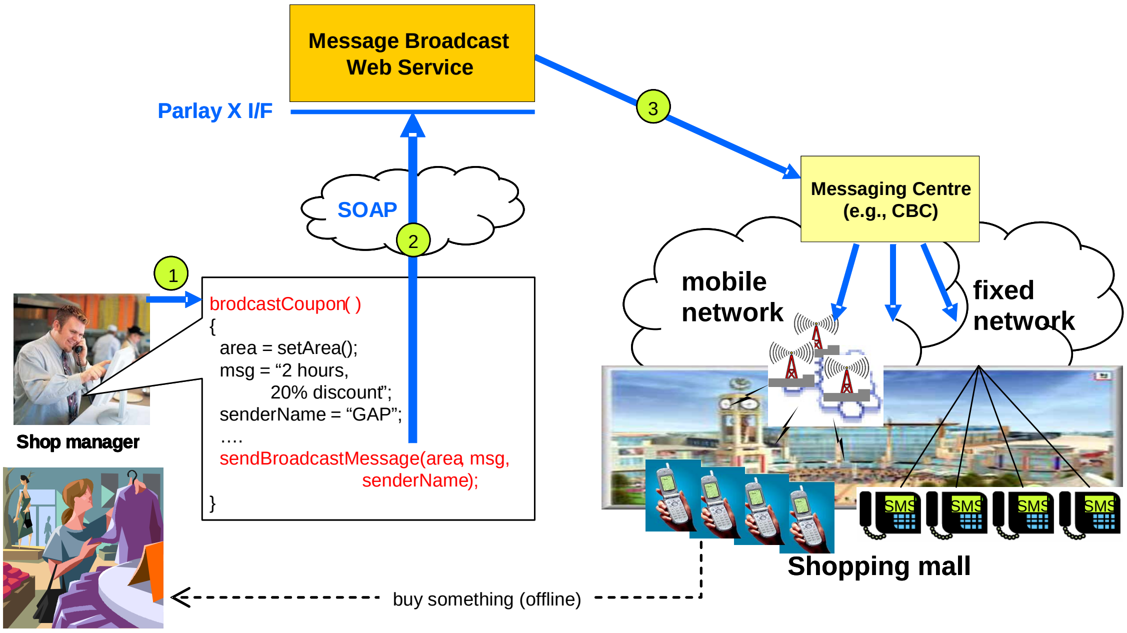
Currently, most of the existing Parlay/Parlay X APIs are defined and used for the point-to-point communication such as TPC, SMS, MMS, etc. But, in some other cases, point-to-multipoint communications - i.e. broadcasting of messages to everyone who is in certain areas or multicasting of messages to a certain group - could be required and considered as very useful and efficient communication method.

Message broadcast is a functionality that allows an application to send messages to all the fixed or mobile terminals in a specific geographical area.

Scenarios:

There are various use cases of using Message Broadcast Web Service including the commercial application. This Web Service could be also used for non-commercial purposes as follows:

- To provide area-based public information such as weather, traffic and other information of common interest.
- To provide emergency information such as severe weather warnings (e.g. typhoon, tsunami), environmental hazards (e.g. chemical spills) and terrorism alerts.



```
brodcastCoupon( )
{
  area = setArea();
  msg = “2 hours,
         20% discount”;
  senderName = “GAP”;
  ….
  sendBroadcastMessage(area, msg,
                       senderName);
}
```

Figure 2

Figure 2 shows an advertising scenario using the Message Broadcast Web Service to broadcast messages describing shop discount offers inside, and in the vicinity of, a shopping mall. A shop manager who wants to increase sales during a holiday period can make use of a message broadcast application. By using the application, the manager can set the targeted area, compose the sales message and identify the shop offering the discount (1). Then, the application uses the Parlay X interface to invoke the Message Broadcast Web Service operation (2). After invocation, the Message Broadcast Web Service sends a message delivery operation to the messaging centre, e.g. the CBC (3). Subsequently, the shop discount message is delivered to all the terminals within the targeted area.

Requirements Description:

- Broadcast a message to a designated area with frequency and interval.
- Retrieve the delivery status of previous broadcast request.
- Cancel the previous broadcast request.
- Be notified when the delivery has been done or is impossible.

Proposed Solution and Further Considerations:

This function is viewed as appropriate for implementation at an abstraction level consistent with that of the Parlay X Web Services. This function is also supported in the underlying network capabilities, as described in the following specifications:

- TS 123 041 [6].
- TS 123 032 [7].
- SMPP v5.0 from SMS Forum.

6.2 Multimedia Stream Control

Issues and Motivation:

New terminals are advanced and can now handle streaming media like video clips and music. Operators have invested in Streaming Servers and a new API is needed to support third party access to these Servers to stream content to terminals.

An alternative is to download all content to the operator and let subscribers access it via the operator, i.e. the walled garden approach. Another alternative is to open the network completely. Neither of these are viable options.

The service provided to an end-user is consumption of streaming multi media. The end-user has a terminal that is able to request a media stream, either from a built-in player, or an installed application. The terminal can be any terminal with streaming media playing capabilities, and the service should allow a user to transfer between his terminals.

Scenarios:

- The basic scenario is where an individual is browsing the Internet and finds some interesting content that he/she wants to watch. The end-user is then either doing this through the operator's portal or accesses the content provider's site. In the first case, the request is then processed through the portal, and charged as the stream is set-up. In the second case, the content provider redirects the request to the operator, so that the terminal capabilities can be collected and charging is done, before the stream is started.

- One scenario is an individual watching his favourite sports stream on his TV or PC at home, but he/she must leave the house of some reason, and still wants to continue the session on his mobile terminal. In that case he/she would transfer the ongoing session to the other terminal with other capabilities, since he already paid for the entire episode.

- The other way is a similar case where an individual arrives at a destination with better viewing capabilities.

Requirements Description:

The following use case diagram defines the core functionality which should be supported by the MultimediaStreamingControl web service. These high level logical use cases will be supported by the defined API, but not necessarily on a one to one mapping basis.

Proposed Solution and Further Considerations:

This function is viewed as appropriate for implementation at an abstraction level consistent with that of the Parlay X Web Services.

Figure 3 shows streaming content delivered to mobile terminals, but is not restricted to any particular terminal type. The content is either on a location accessible over the Internet, or stored locally on a operators domain or the content providers domain. The access to the content is done through a smart router controlling the stream towards the terminal. Transcoding of a stream is a optional feature, that could allow users to switch between different terminal and networks, while consuming content from a streaming source.

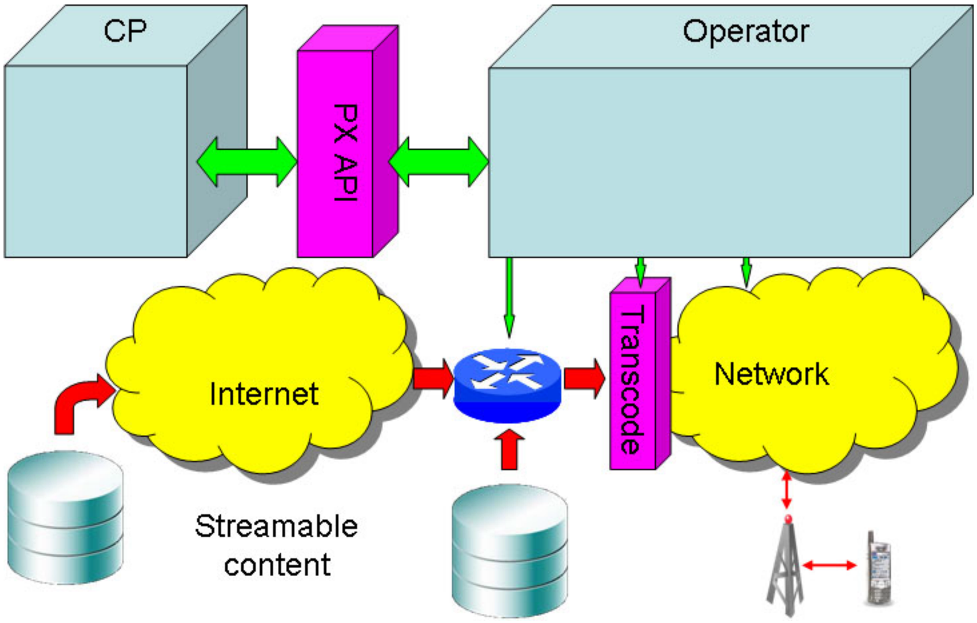


Figure 3

6.3 Extend mobility to include Geocoding

Issues and Motivation:

While the Parlay X Terminal Location Web Service provides access to the geographical coordinates at which a terminal is located, the Geocoding Web Service provides an additional level of refinement, allowing the service developer to work with actual location addresses.

Scenarios:

What is(are) the current physical location address(es) of my friend (family)?

Are any of my friends also shopping or dining in the mall?

Requirements Description:

Services are specified to enable an application to:

- Request the location address of a terminal number.
- Request the location addresses of a group of terminals.
- Request the terminal number(s) known to be at the specific location address.

Proposed Solution and Further Considerations:

This function is viewed as appropriate for implementation at an abstraction level consistent with that of the Parlay X Web Services.

The figure below shows the Geocoding Web Service architecture comprising Parlay X Application, Parlay X Gateway, Parlay Gateway, GMLC/MPC, Addressing server. OMA/MLP is used between the Parlay X Gateway and GMLP/MPC and OpenLS/XML is used between the Parlay X Gateway and the Addressing server.

If the subscriber asks for someone's address or for group addresses, the Parlay X Application invokes the Geocoding Web Service. The Parlay X gateway gets the geographical coordinates of the terminal from GMLC/MPC and then feeds the retrieved geographical coordinates to addressing server. Finally it obtains the location address.

If the subscriber asks for the terminal number(s) at a specific address, the Parlay X Application invokes the Geocoding Web Service. The Parlay X gateway gets the geographical coordinates of address from an addressing server, and then feeds the retrieved geographical coordinates to GMLC/MPC. Finally it gets the terminal number(s) for that location.

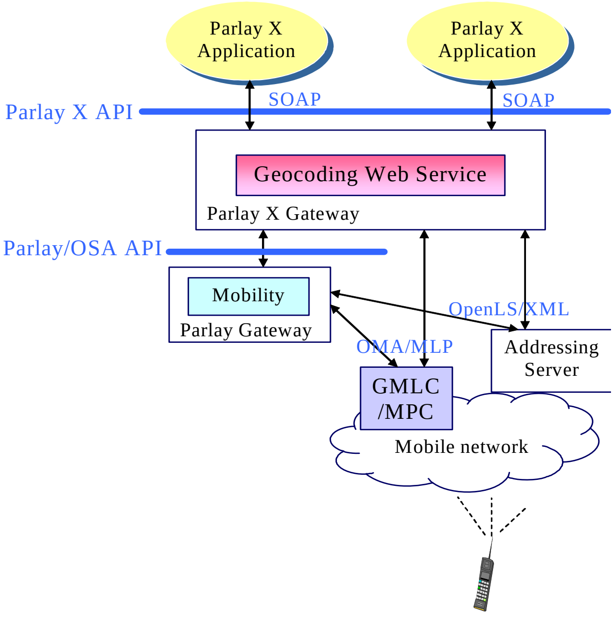


Figure 4

6.4 Service Brokering

Issues and Motivation:

OSA Service Brokering support requires API level capabilities like Service Selection, Service Provisioning, Feature Interaction and Service Chaining. The concept of Service brokering in this context is the ability to package, provision and supply a set of applications or services onwards to the application server implementing the business logic that requires the use of such a service broker functionality within an OSA environment.

Service broker function shall enable the delivery of multiple services in an operator network in a managed and controlled fashion. Therefore whenever an event occurs, there is a need to ensure that the set of applications or services that may act upon that event are invoked in a manner that does not conflict with any other application or service defined in the provisioned package of applications or services.

Scenarios:

- A network event such as a call trigger may result in the need to resolve conflicts between different OSA applications and related service delivery platforms.
- A OSA SCS may receive or generate an event that requires the use of further OSA SCSs, for example Policy Management, Charging etc., transparent to the application using the SCS.
- A OSA SCS may generate an event that may result in the need to resolve conflicts between multiple OSA applications.

Requirements Description:

Provide a OSA Service Brokering API capable of supporting the following features:

- Provisioning and Management of all data necessary to support service brokering.
- Evaluation of service brokering data to control execution of service scenarios.
- Transparent of service brokering location, include support for network service brokering, OSA SCS service brokering and OSA application service brokering.

Proposed Solution and Further Considerations:

The service broker function enables the brokering of multiple services in a managed and controlled fashion.

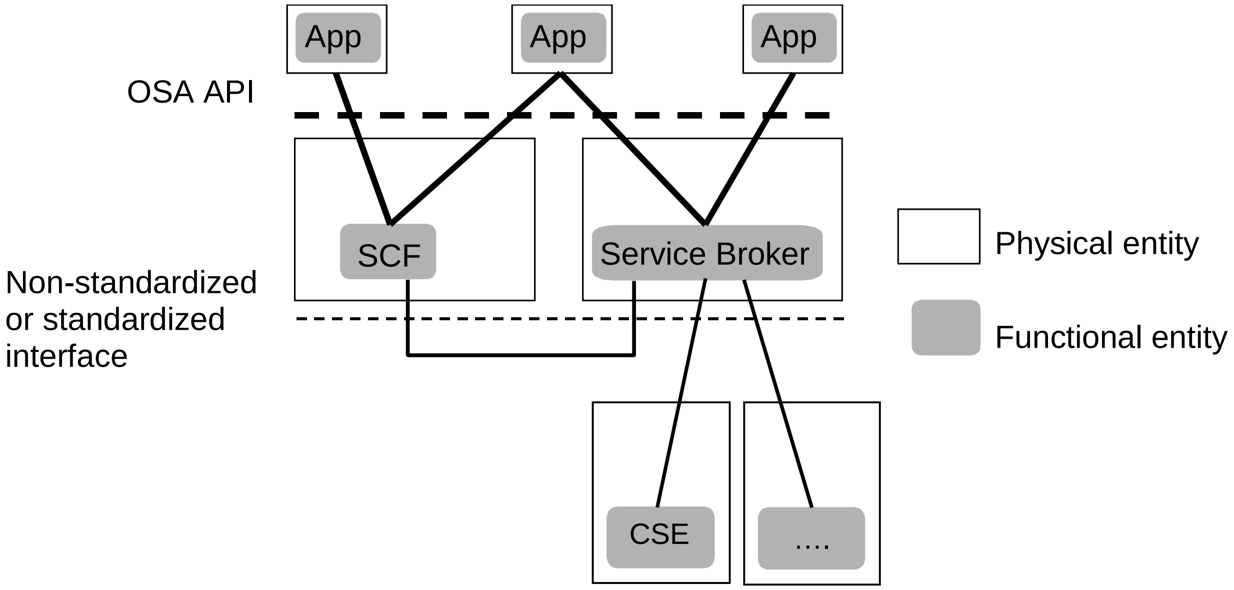


Figure 5: Service Broker function implemented as separate physical entity

Figure 5 depicts the interfaces between the various entities requiring brokering of services within the OSA environment. In this diagram the service broker is identified as an independent physical entity to that which supports the SCF. This is not mandated, but is depicted to indicate that such an architecture is possible.

The OSA Service Broker API includes the ability to specify provisioning and configuration data required to control the service brokering.

The messaging interfaces between Service Broker and OSA SCSs or other service delivery nodes/elements is based upon the network interfaces supported by each system. Note that the above architecture does not preclude the OSA SCS communicating directly with network entities where service interaction is not required.

6.5 QoS for end-user/s involved in an application session

Issues and Motivation:

In today's networks we have resources that provide us with high-speed information flows. The ability to make use of QoS here should enhance the end-users experience and provide them with a more reliable service, increasing the number of services available and inevitably, the **revenue** earned from such networks.

The end-to-end QoS received by the application will directly translate into a user perceived worth of the new applications and related services, thereby giving the ability to individualize and enhance the subscriber experience. It will also provide the application provider the ability to tailor services with a differentiated QoS; one service with one or more QoS classes thereby reaching a wider audience.

Scenarios:

Not applicable.

Requirements Description:

The OSA interface shall provide functionality to enable an application to dynamically change the quality of service available on the end user connection:

- An application may request to change the default quality of service available on the end user's connection.
- An application may request to change the quality of service available on the end user's connection on a temporary basis.
- Applications are able to view transaction history and current quality of service status for reasons such as self-care.

QoS features are defined within the network and shared offline beforehand with application providers, with a clear indication as to which of these can be used as default QoS features and temporary QoS features.

Proposed Solution and Further Considerations:

This function is viewed as appropriate for implementation at an abstraction level consistent with that of the Parlay X Web Services.

"Application Driven QoS Service" (ADQ) is a new service specification that will enable Applications developed by Application Providers to dynamically change the quality of service available on the end user connection. Quality of service can be dynamically changed by setting up QoS features on end user connections. This service is enabled through the use of a Web Service interface, as illustrated in figure 6.

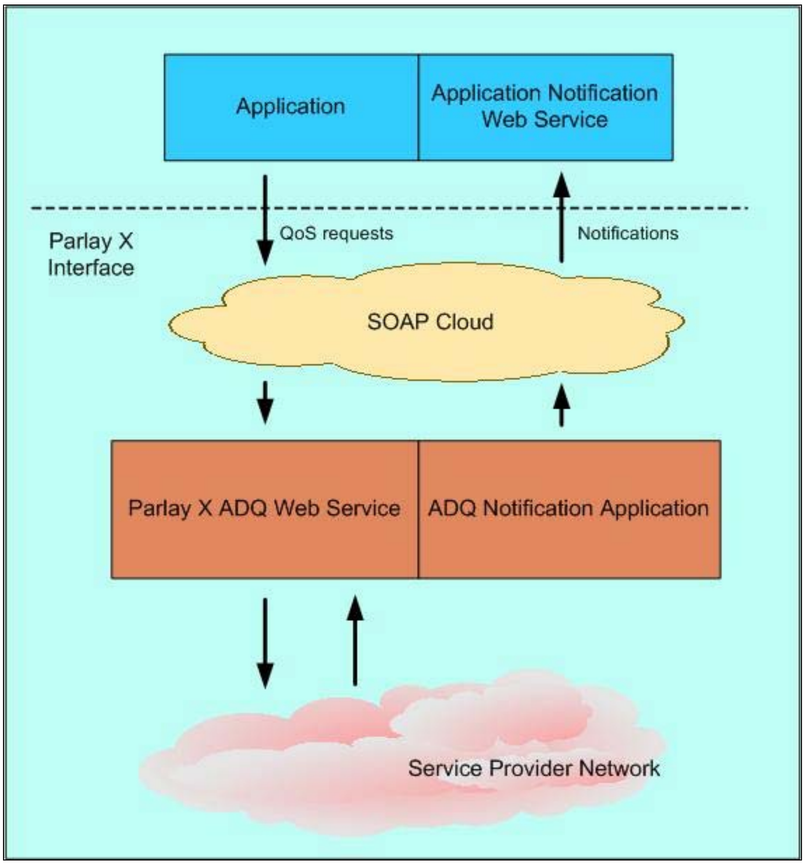


Figure 6

6.6 Multimedia Multicast Control

Issues and Motivation:

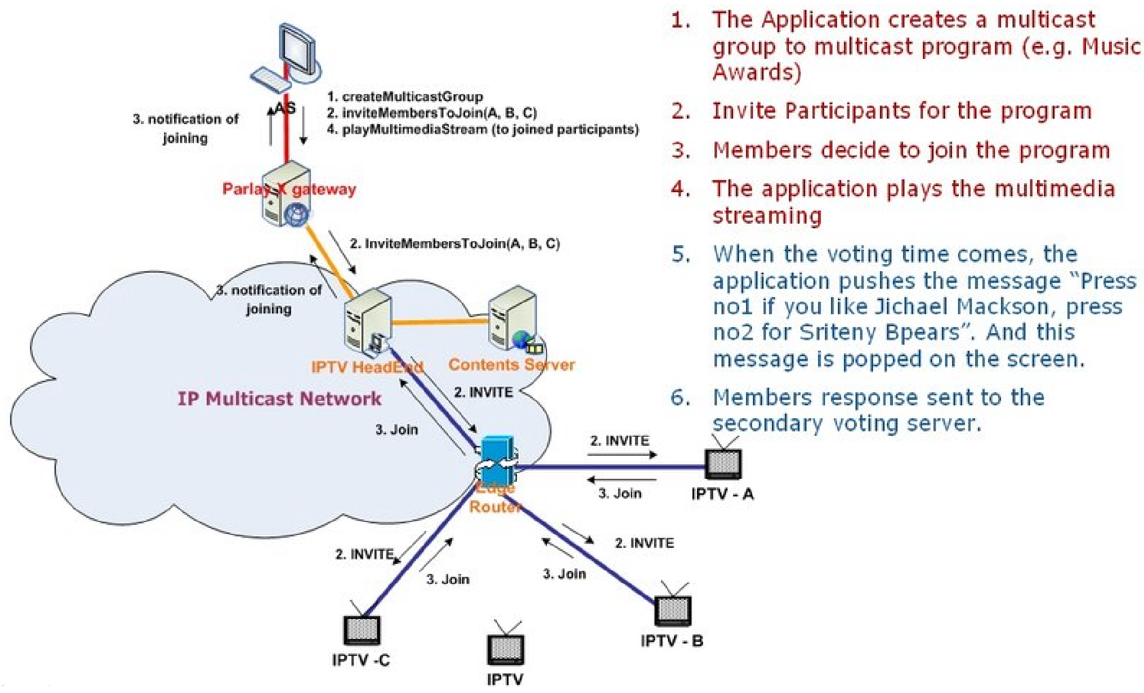
Most voice and data service providers are deploying or planning to deploy converged services, such as IPTV and other emerging services, which combine communication and broadcasting features. Using IPTV, the service provider can offer various value-added services such as Caller ID.

A Caller ID service enables a subscriber, who is watching TV, to receive a voice call identification on the screen, and be provided with the option to pause and record the video stream for the duration of the voice call.

Service implementation requires the subscriber to belong to a multicast group, whose membership and capabilities could be managed using a Parlay API.

Scenarios:

The IPTV service is one of the IP multicast based services. The service scenario and underlying network architecture is depicted in figure 7:



1. The Application creates a multicast group to multicast program (e.g. Music Awards)
2. Invite Participants for the program
3. Members decide to join the program
4. The application plays the multimedia streaming
5. When the voting time comes, the application pushes the message "Press no1 if you like Jichael Mackson, press no2 for Sriteny Bpears". And this message is popped on the screen.
6. Members response sent to the secondary voting server.

Figure 7

Requirements Description:

Permit a third party application to perform:

- Multicast Session Management: create session, delete session, get session info.
- Multicast Member Management: invite participant to join or leave.
- Multicast Stream Management: allowing the application to control and manipulate(play/pause/resume/stop) a multimedia stream on a multicast session.

Proposed Solution and Further Considerations:

This function is viewed as appropriate for implementation at an abstraction level consistent with that of the Parlay X Web Services.

Figure 8 shows the proposed architecture and the underlying networks. The Multimedia Multicast Control Web Service can communicate with the BM-SC in GPRS packet network or the head-end in IP multicast network using a proprietary protocol.

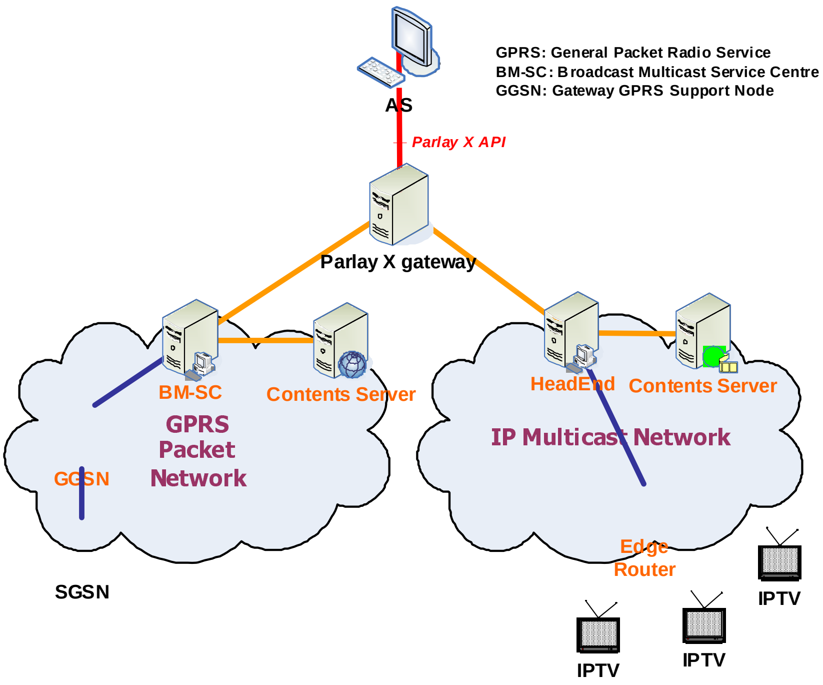


Figure 8

6.7 Device Capabilities and Configuration

Issues and Motivation:

Desire to allow applications to remotely install and configure new services on the user terminal. Newly purchased services are readily available without any action required by the customer.

Scenarios:

Use case 1: A customer service representative application pushes a new GPRS configuration to a customer complaining about data connection issues.

Use case 2: A customer purchases a new device. The default software on the phone is checked and updated if not up-to-date. Specific application software associated with the device phone number is then pushed to the device and installed. If an application associated with the phone number is not supported on the new device, an exception triggers the generation of an SMS message to the device instructing the user to contact customer care.

Use case 3: Operations reviews detect poor performance of an application on a specific Device_ID. Software updates are pushed to all users with the associated application_ID AND Device_ID.

Use case 4: A carrier wishes to push software to all devices with a specific Device_ID.

Use case 5: A user installs his/her SIM card into a different terminal (this may be a terminal with the same Device_ID or a different Device_ID). The applications associated with the phone number are pushed to the terminal. If an application associated with the phone number is not supported on the new terminal, an exception triggers the generation of an SMS message to the device instructing the user to contact customer care.

Use case 6: The user calls the Service Provider to cancel a subscription to a client application. The application is removed from the Device.

Use case 7: The user acquires a second hand device. The user wishes to remove previously installed client applications that are not subscribed to. The user initiates an application (SyncApplications) on the Device which retrieves a list of subscribed Application_IDs and checks the installed applications against this list. If an unsubscribed application exists on the Device, the user is asked:

- if he/she wishes to subscribe to the application/service;
- if the user chose to subscribe to the application/service, the user is sent an SMS message with instructions on invoking a subscription;
- if the user chose not to subscribe to the application/service the user is asked if he/she wants to remove the application from the device. If the user chooses to remove the application from the device, the application is removed.

Requirements Description:

- Retrieve Device Capabilities such as the device identifier and a URL to a file containing a user profile XML file.
- Device Change Notification.
- Device Configuration:
    - pushConfiguration;
    - getConfigurationList;
    - getConfigurationHistory.

Proposed Solution and Further Considerations:

This function is viewed as appropriate for implementation at an abstraction level consistent with that of the Parlay X Web Services.

The Parlay X web service relies on a Device Management Solution (DMS) to implement the functionalities of the present document, as depicted in figure 9.

The Parlay X or DMS have to include a repository of device capabilities in order to make a correspondence between a subscriber address and his/her device capabilities. The DMS has to include a configuration file repository. These files should respect the OMA Client Provisioning standard. Usually sent to the subscriber device by SMS messages these files may configure settings such as WAP, MMS, Emails, etc. The DMS has to include an application inventory. Existing DMS are using different methods of application storage and description, but most of them are using syncML to exchange information with the subscriber device. SyncML is an OMA Standard for Device Management.

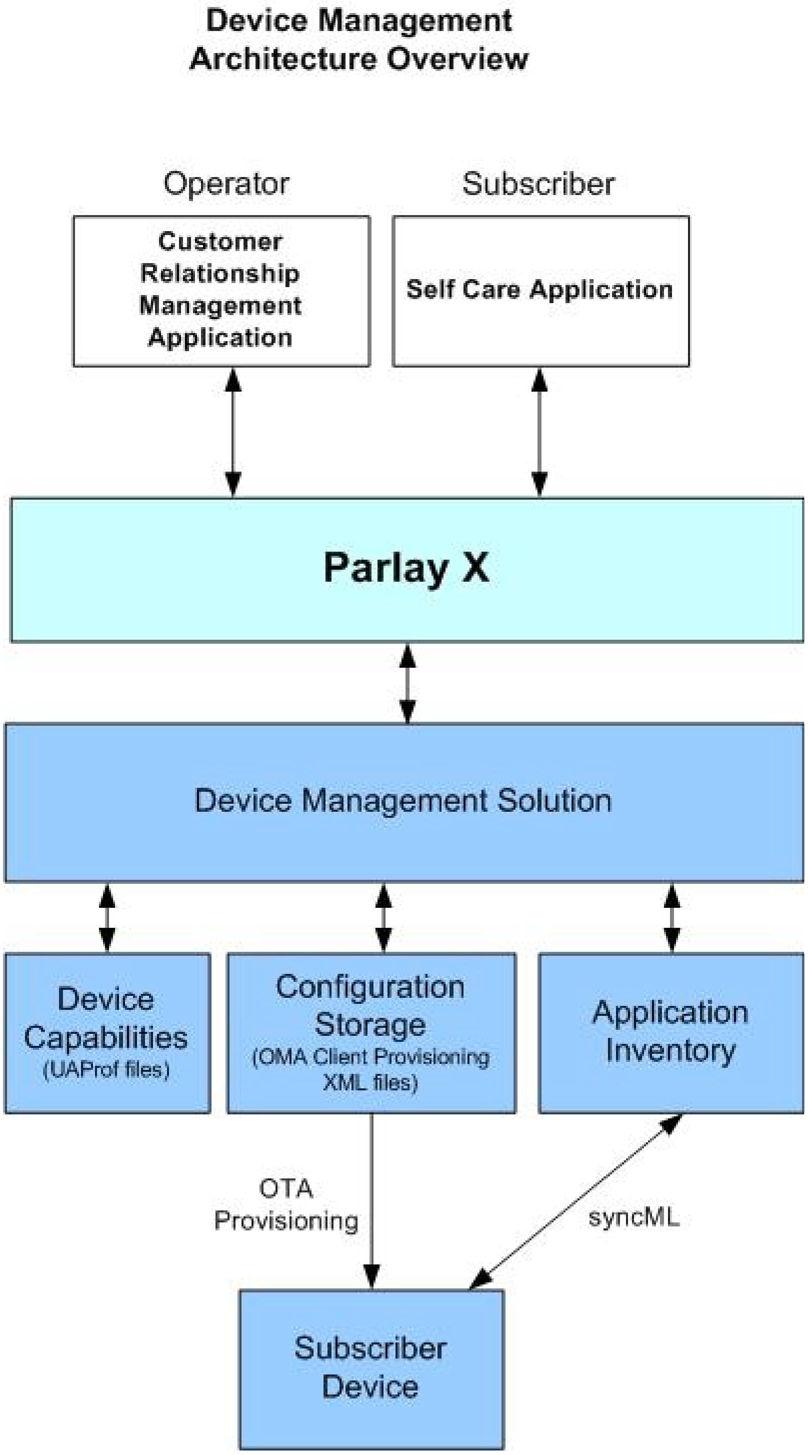

Device Management
Architecture Overview
Operator
Subscriber
Customer Relationship Management Application
Self Care Application
Parlay X
Device Management Solution
Device Capabilities
(UAProf files)
Configuration Storage
(OMA Client Provisioning XML files)
Application Inventory
OTA Provisioning
syncML
Subscriber Device

Figure 9

Annex A (informative): Bibliography

- ETSI TS 123 198: "Universal Mobile Telecommunications System (UMTS); Open Service Access (OSA); Stage 2 (3GPP TS 23.198)".
- ETSI TS 123 127: "Universal Mobile Telecommunications System (UMTS); Virtual Home Environment (VHE)/Open Service Access (OSA) (3GPP TS 23.127)".

History

Document history		
V1.1.1	December 2007	Membership Approval Procedure MV 20080208: 2007-12-11 to 2008-02-08
V1.1.1	February 2008	Publication