ETSI EG 201 988-3 V1.1.1 (2005-07)

ETSI Guide

Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Service Provider Access; Open Service Access for API requirements; Part 3: Version 3

Reference

DEG/TISPAN-02009-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

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

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 3 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".

Introduction

The present document contains the Requirements capture for ETSI 3.0 "Third Party API" protocol specification ES 203 915 series [1].

1 Scope

The present document contains the functional requirements for Open Service Access API Requirements Version 3.0. The present document has been compiled in conjunction with Parlay and represents the fifth phase of the Parlay API. The ETSI and Parlay API have been specified and designed using the requirements identified in the present document. 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 2.0 API and that of the Parlay 4.0 specification and should be fully backward compatible. This means that any network operator implementing ETSI Phase 3.0 or Parlay 5.0 should be able to interwork with a client application provider implementing ETSI Phase 2.0 or Parlay 4.0. In other words ETSI Phase 3.0 and Parlay 5.0 will retain ETSI Phase 2.0 and Parlay 4.0 as a complete subset.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- 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.
- For a non-specific reference, the latest version applies.

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

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

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

3 Abbreviations

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

API	Application Program Interface
ASP	Application Service Provider
IP	Internet Protocol
ISDN	Integrated Service Digital Network
SS7	Signalling System 7
UI	User Interaction

4 ETSI 3.0/Phase 5 Parlay 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 Telephony 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 centres, virtual private networks and fully interactive multimedia.

The API provides the common interfaces to a variety of services. For the services to work together in a coherent fashion, "framework" functions are required and are also included in the present document.

The remainder of the present document captures the requirements for the ETSI Phase 3.0 and Parlay 5.0 specification.

4.1 Framework interface and Service Interface

Services and the framework functionality will be exposed via interfaces. These interfaces will be called the service interface and framework interface respectively.

5 Proposed enhancements to existing Interfaces

5.1 General requirements

5.1.1 Backwards Compatibility/Deprecation

Source: Parlay

Issues & Motivation:

It needs to be considered what can be done if we find that certain interfaces in Release 4.0 are found to be unstable and therefore require appropriate modification. If we use the concept of deprecation then we can effectively provide new methods to that interface where the old methods are incorrect. This means that the two methods will exist side by side in the same interface for the same purpose in 5.0. One method may not be complete but the other is! The methods that are incorrect would be removed in further versions of the API.

Requirement Description:

The Parlay 5.0/OSA 3.0/ETSI SPAN 3.0 APIs shall be backwards compatible. This has two aspects:

- A client application utilizing Parlay 4.0/OSA 2.0/ETSI SPAN 2.0 APIs shall run without change (not even re-compilation) against a server providing Parlay 5.0/OSA 3.0/ETSI SPAN 3.0 APIs.
- A deprecation mechanism shall be defined that allows the removal of outdated methods or interfaces in a well-defined, step-wise approach.

5.2 Framework

5.2.1 Federation of Frameworks

Source: Telecom Italia (Durango SA1 meeting S1- 021585)

Issues & Motivation:

The federation function enables that SCFs registered on one framework could be accessed also by the applications authenticated by federated frameworks, and that an application could use SCFs registered on different federated frameworks.

NOTE: This capability is already available using the existing API. The text remains within the Requirements document for completeness.

To specify the federation functional requirements, the following terms are introduced:

- **federated frameworks:** frameworks in different administrative domains that agreed on the set up of a federation agreement;
- **donor framework:** it is a federated framework that makes a specific SCF visible to the corresponding one or more federated frameworks (the receiver framework);
- **receiver framework:** it is a federated framework that can access a specific SCF offered by a corresponding federated framework (the donor framework);
- **federated SCF:** it is an SCF provided by a donor framework and accessed by a receiver framework.

The federation function enables a "donor" framework to limit the access to the registered SCFs from the federated frameworks, i.e.:

- a "donor" framework shall be able to discriminate the set of SCFs visible to the federated frameworks;
- a "donor" framework shall be able to set some service properties to constraints the usage of its SCFs by the applications authenticated by a federated framework.

Federation function enables a "donor" framework to maintain the full control of its registered SCFs, i.e. handling the integrity management of the SCFs, controlling the access to the SCFs, etc.

Federation function enables the exchange of the information among the federated frameworks concerning the SCFs availability and the associated service properties.

An SCF shall not be aware of the fact that is a "federated" SCF and that may be accessed by applications authenticated by different federated frameworks: therefore, the SCF does not have to support additional functions to handle federation.

An application shall not be aware that it is interacting with SCFs registered on different federated frameworks.

6 New interfaces and areas of involvement

6.1 Policy Management

Source: 3GPP S1-021721 and S1-021722 (Durango)

Issues & Motivation:

Applications shall have the ability to interact with policy-enabled Service Capability Features in a secure manner. The network policies always take precedence over the application defined policies.

The OSA interface shall provide sufficient capabilities to enable applications to request:

To manage the application's policy-related information:

- This allows applications to create, modify and delete policies, policy events and to activate, deactivate and modify policy rules. Policy rules may be expressed with simple data types (such as integers or string) or more complex data types (such as Boolean values, time, lists, meta-variables, etc.). Expression of policy rules shall take into account these complex data types as well as allow for a feature rich set of operands and allow for ability to define user specific functions.

To manage policy event notification:

- This allows applications to register for specific policy events. Once registered for such events, the application shall receive notification of the events until it explicitly requests the termination of the notification request.

To collect policy statistics:

- This allows an application to collect policy related statistics from the network. Examples include success or failure of operations on policies and time stamps of policy events.

To request policy evaluation:

- This allows an application to request that a set of policies is evaluated by the network.

6.2 Multi-media Messaging function

Source: S1-021854 Durango

Issues & Motivation:

The Multimedia Messaging function enables applications to receive and send multi-media messages.

The Multi Media Messaging function should allow applications to:

- send and receive messages both within and outside the context of a session (for session-based and single-shot messaging respectively);
- put messages in the mailbox for storage or for sending by the messaging system (with a copy in the mailbox);
- cancel a message previously sent or query the status of a message previously sent;
- manipulate folders and messages in the mailbox (e.g. copy, move, delete);
- list messages in the mailbox and retrieve complete messages, message headers, message body or parts of the message body.

6.3 User location

Source: S1-021716 Durango (inclusion of LCS enhanced User privacy)

Issues & Motivation:

The User Location functions provide an application with information concerning the user's location.

The user location information contains the following attributes:

- **location** (e.g. in terms of universal latitude and longitude co-ordinates);
- **accuracy** (value depending on local regulatory requirements and level of support in serving/home networks; note that the accuracy of the serving network might differ from that in the home environment);
- **age** of location information (last known date/time made available in GMT).

The following functions shall be provided:

- report of location information:
    - the application shall be able to request user location information;
    - by default the location information is provided once; the application may also request periodic location reporting (i.e. multiple reports spread over a period of time).

- notification of location update:
    - the application shall be able to request to be notified when the user's location changes, i.e. when:
        - the user enters or leaves a specified geographic area;
        - the user's location changes more than a specified lower boundary. The lower boundary can be selected from the options provided by the network.

The application shall be able for each user to start/stop receipt of notifications and to modify the required accuracy by selecting another option from the network provided options.

- Access control to location information:
    - the user shall be able to restrict/allow access to the location information. The restriction can be overridden by the network operator when appropriate (e.g. emergency calls).

6.4 Text to Speech and Speech to Text functions

Source: N5 - 021045 CN5 #21 Dublin

Issues & Motivation:

Currently the Parlay user interaction (UI) interfaces lie behind the state of the art in terms of controlling media resources. The advent of Voice XML has advanced UI functionality beyond that available through the Parlay API.

In order to improve the levels of voice interaction possible within a current Parlay system the developer must abandon the Parlay User Interaction interfaces and implement a direct interface between their application logic and a Voice XML server. In addition they must provision VoiceXML scripts onto a web server (either as part of the Parlay App Server or separate) in order to perform their interaction. This also requires the Parlay developer to learn VoiceXML.

This work is being performed to understand whether there is a role for a Parlay Gateway to offer a mapping and control layer in order that the developer may deploy Parlay applications that offer an improved level of Voice Interaction but allowing voice interaction control through the Parlay UI API.

There are two main issues to understand:

1) Can the Parlay UI API support interaction with Voice XML servers in such a way that the functionality offered to the Parlay developer is improved but at a level, which is still abstracted away from the VoiceXML specification and in the style of Parlay?

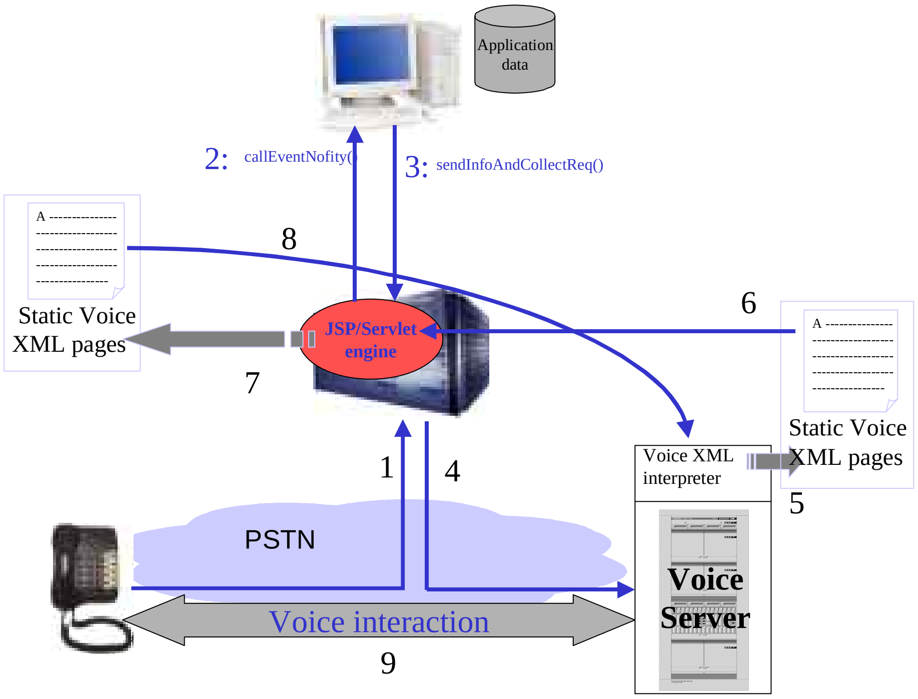
2) Can a Voice XML server be integrated into a Parlay system in such a way that the gateway controls the setting up of a voice connection to underlying interaction systems and thus hides complexity from the developer?

Why not just use Voice XML?

Voice XML offers a powerful environment for interacting with customers via text-to-speech, advanced speech recognition, message recording features and more. Many Voice XML servers also offer telephony capability enabling the transfer of calls and setting up and disconnection of calls.

However the benefit of Parlay is to allow a developer to use a common environment to put together services, which make use of many features. For instance a unified communications service may rely on Parlay Call Control for setting up calls, Parlay Messaging to send a text message notification and Content Charging to check the user has credit to carry out their services.

In addition a Parlay platform has access to these features at the core network level. This gives far more control to the application and means that call costs via Parlay would be significantly cheaper than using an edge of network Voice XML server provided by a Voice Application Service Provider (ASP) who introduces their own commission.



Proposal:

It is proposed that the Parlay User interaction API is modified to include capabilities that enable access to Voice recognition systems.

6.5 Requirements on interfaces at different levels of abstractions

Source: N5 - 021117 CN5 #21 Dublin TI labs

Issues & Motivation:

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. The abstraction levels and the programming formalism should be identified according to the needs of the programmers' communities.

All the interfaces shall be integrated in the context of the OSA architecture:

- they shall guarantee a secure and controlled access to the service capabilities;
- it shall be possible to introduce them in an incremental way;
- they should allow the creation of applications triggered by network events.

The interfaces shall be defined using state of the art specification formalisms (e.g., UML, WSDL), and realized using different distributed processing technologies, including CORBA and Web Services.

6.6 User-Application Authentication functions

Source: TS 122 127 [2] Version 6.7.0 latest text

Issues & Motivation:

The User-Application Authentication functions provide to applications support for authentication of their users. It also provides an "application-specific user identifier" to be used as a parameter in invocation of other OSA Network functions, when requested by the application.

The User-Application Authentication functions shall authenticate a user upon requests of an application; this requires the application to provide as an input the subscriber's credentials, which enable secure method of authentication (e.g. subscriber's certificates).

The User-Application Authentication functions shall return to the invoking application an "application-specific user identifier" (a true identity or alias) that identifies the authenticated user, when requested by the application. The identifier may be used by the application to recognize a user through several accesses to the application; it may also be used by the application as a parameter in invocation of other OSA network functions (e.g. for User Location function).

The User-Application Authentication functions shall support privacy settings defined by the user.

If the subscriber's privacy settings so require, the "application-specific user identifier", returned by User-Application Authentication function to the invoking application, shall be an alias. Otherwise, the "application-specific user identifier" shall be the true identity of the subscriber (e.g. MSISDN).

When the application invokes OSA Network functions related to subscriber (e.g. Location, Presence), the subscriber's identifier shall be included in the request. An application may request it from the User-Application Authentication function.

When an OSA Network function receives the request from the application and the subscriber's identifier is an alias, the OSA Network Function shall invoke the User-Application Authentication function to translate the alias to the subscriber's true identity (e.g. MSISDN).

6.7 User Binding functions

Source: Telcordia/NTT N5-030626 Bangkok Oct03

Issues & Motivation:

The Mobility APIs are said to be designed for mobile, fixed and "IP" networks. However, the mapping to mobile networks is most intuitive, while the mappings to fixed and "IP" networks is limited. In order to create a feature rich services creation environment for (mobile) "IP" networks where not only terminal mobility but also personal mobility is supported, new User Binding notification functions, which enable applications to know UE binding requests, control and make use of them, must be introduced.

The User Binding notification functions shall exploit standard IP Session binding protocols, i.e. SIP REGISTER.

An example use case is as follows:

When an OSA gateway accommodating SIP/H323 terminals receives a Register/ARQ message from a particular terminal, a check procedure should be executed to decide whether or not the terminal sending the message has a contract with the Gateway provider, ex. usage of the operators SIP/H323 network services. Applications associated with these subscribers will then receive an event from the gateway and check the subscribers' data and then pass the check results to the Gateway.

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	May 2005	Membership Approval Procedure MV 20050701: 2005-05-03 to 2005-07-01
V1.1.1	July 2005	Publication