SLOVENSKI STANDARD
SIST-V ETSI/EG 201 988-2 V1.1.1:2003

01-november-2003

Storitve in protokoli za napredna omrežja (SPAN) - Zahteve dostopa ponudnika storitve (SPAR) - Odprt dostop do storitve za zahteve API - 2. del: Različica 2

Services and Protocols for Advanced Networks (SPAN) - Service Provider Access Requirements (SPAR) - Open Service Access for API requirements - Part 2: Version 2

Ta slovenski standard je istoveten z: EG 201 988-2 Version 1.1.1

ICS:

33.040.35 Telefonska omrežja Telephone networks

SIST-V ETSI/EG 201 988-2 V1.1.1:2003 en

ETSI EG 201 988-2 V1.1.1 (2003-04)

ETSI Guide

Services and Protocols for Advanced Networks (SPAN); Service Provider Access Requirements (SPAR); Open Service Access for API requirements; Part 2: Version 2

Reference
DEG/SPAN-141606-2

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, send your comment to:
editor@etsi.org

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).

All published ETSI deliverables shall include information which directs the reader to the above source of information.

Foreword

This ETSI Guide (EG) has been produced by ETSI Technical Committee Services and Protocols for Advanced Networks (SPAN).

The present document is part 2 of a multi-part deliverable covering Service Provider Access Requirements (SPAR); 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 Version 2 Open Service Access API protocol specification.

1 Scope

The present document contains the functional requirements for the second phase of the ETSI Open Service Access API, as defined in ES 202 915 [4]. The present document has been compiled in conjunction with Parlay and represents the fourth phase of the Parlay API. The ETSI and Parlay API has been specified and designed using the requirements identified in the present document. The requirements are intended to provide the necessary functionality for benchmark applications.

The new requirements build upon the ETSI OSA Phase 1 API in ES 201 915 [3] and the Parlay 3 specification and should be fully backward compatible. This means that any network operator implementing ETSI OSA Phase 2 or Parlay 4 should be able to interwork with a client application provider implementing ETSI OSA Phase 1 or Parlay 3. In other words ETSI OSA Phase 2 and Parlay 4 will retain ETSI OSA Phase 1 and Parlay 3 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 TS 129 198: "Universal Mobile Telecommunications System (UMTS); Open Service Access (OSA) Application Programming Interface (API)".

[2] ETSI TS 123 127: "Universal Mobile Telecommunications System (UMTS); Virtual Home Environment (VHE) / Open Service Access (OSA); Stage 2 (3GPP TS 23.127 version 5.2.0 Release 5)".

[3] ETSI ES 201 915: "Open Service Access (OSA); Application Programming Interface (API)".

[4] ETSI ES 202 915: "Open Service Access (OSA); Application Programming Interface (API)".

3 Abbreviations

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

API	Application Program Interface
ASP	Advanced Signal Processor
CC	Call Control
DSC	Data Switch Control
ETS	Emergency Telecommunications Service
HTTP	Hyper Text Transfer Protocol
IP	Internet Protocol
LIF	Location Interoperability Forum
MIS	Management Information System
MPCC	Multi Protocol Call Control
OSA	Open Services Access
PDP	Packet data protocol
SIP	Session Initiation Protocol
SCS	Service Capability Server
SOAP	Simple Object Access Protocol

SS7	Signalling System 7
USSD	Unstructured SS Data
W3C	World Wide Web Consortium
XML	eXtended Markup Language

4 ETSI OSA Phase2/Parlay Phase 4 API domains

The ETSI/Parlay API is an open, technology-independent, and extensible interface into networking technologies. The 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 multi-media 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 multi-media.

The rest of the present document is structured to capture all of the requirements that are deemed necessary to enhance the existing ETSI OSA Phase 1 and Parlay 3.2 specification to an ETSI OSA Phase 2/Parlay 4.0 status.

4.1 Framework interface and service interface

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.

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

Issue/Motivation:

It needs to be considered what can be done if we find that certain interfaces in Parlay 3.1 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 Parlay 4.0. 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 4.0/OSA 2.0/ETSI SPAN 2.0 APIs shall be backwards compatible. This has two aspects:

- A client application utilizing Parlay 3.0/OSA 1.0/ETSI SPAN 1.0 APIs shall run without change (not even re-compilation) against a server providing Parlay 4.0/OSA 2.0/ETSI SPAN 2.0 APIs.
- A deprecation mechanism shall be defined that allows to remove outdated methods or interfaces in a well-defined, step-wise approach.

5.1.2 Emergency preparedness

Source: Telcordia

Issue/Motivation:

There is a need to extend A/IN-based facilities defined for national emergency calls to Next Generation Networks and APIs. The U.S. Government and other countries have sponsored programs over the past 15 years to ensure, via standards and implementation programs, that National Emergency calls enjoy priority handling, Network Management Control exception and Alternate Carrier Routing, etc. This initiative is known as Emergency Telecommunications Service (ETS). Although this is currently available for voice calls only there is also a need to handle new types of communication (data, email, video, multi-media), new types of networks (wireless, packet) and technology (protocols, architectures). These requirements impact Call Control and Policy Management and may also impact Mobility, Charging and Framework (e.g. for location-based service, accounting bypass, security, etc.).

Requirement description:

The Parlay/OSA/ETSI SPAN APIs shall support the Emergency Telecommunications Service. In particular, they shall provide client applications with means to make use of the Emergency Telephony Service.

Possible solution and further considerations:

To support the Emergency Telecommunications Service, an optional parameter could be provided within the call control and other interfaces (e.g. CC, MPCC, DSC).

5.1.3 Balancing up of interfaces

Source: Eurescom

Issue/Motivation:

Many of the Parlay/OSA/ETSI SPAN APIs are highly asymmetric between application and gateway. As the capabilities of terminals connected to networks continues to grow, network based applications will require greater awareness of these capabilities. Likewise, as new network protocols such as SIP, which are peer-to-peer in nature, are developed, new types of functionality and control will become possible. The "Balancing up" of interfaces is concerned with identifying areas where the asymmetry of Parlay may cause limitation in functionality or feature interaction problems.

Although many of these asymmetries are particularly apparent on SIP networks, many aspects identified will not be restricted to SIP.

The work is aimed at identifying and suggesting changes to Parlay to help in solving feature interaction problems and to support fully the flexibility which new networks, protocols and terminals enable.

Scenarios:

This clause provides scenarios and examples of situations where more balanced interfaces might be beneficial.

At present the work is expected to have a broad impact across the service interfaces, for example.

Call Control

An application can create a new call leg within a call, but a network cannot create a call leg and notify an application that a call leg has been created. Terminals and protocols which allow a client to add an additional party to the call should be able to do so, making the application aware of the new party.

An application can attach and detach call legs from a call, but the application cannot be notified that the call legs have been attached or detached by the connected party. This causes problems because if a caller detaches from a call, then the application is not aware of this. In a symmetrical model, the media stream could be connected or disconnected by either application or client, and the application could be notified of changes.

Mobility interfaces

The mobility interfaces allow an application to query the status and location of an address. However, an application cannot request notification of incoming requests for the status or location of address and return a response. A scenario where this causes a problem is where a unified communications application provides a "one-number" service, and a second application requests the status or location of the user. The unified communications application must have be able to receive status requests and respond to them.

Requirements description:

- To identify aspects of Parlay where asymmetries can cause limitations. This includes call control, mobility, terminal capabilities and user interaction interfaces.
- To collate the information from the comparisons and to identify service scenarios where the existing interfaces are too restrictive and where symmetrical behaviour would be advantageous.
- To ensure that any symmetrical interfaces proposed do not compromise compatibility with existing asymmetric networks.

Possible solution and further consideration:

Modified Parlay interface class diagrams, interface definitions and data types or recommendations via reports.

5.2 Framework

5.2.1 Framework information model

Source: Eurescom

Issue/Motivation:

An Information Model for the Parlay/OSA Framework (FIM, Framework Information Model) can answer to several needs, coming from different actors in the possible business models, but the first reason to define such a model is to have an agreed, common view and a "common language" to address the same concepts when analysis related to complex data structures, relationships and objects, have to be done. Since work done so far, both on standard and on the vendors' side, indicates that the Framework functionality is the heart of the Parlay Gateway system, the availability of a Framework Information Model appears more and more relevant.

The needs of new interfaces (e.g. user profile service interface), or to improve the existing ones, makes the FIM definition relevant also on the standard side. In particular, a clear definition of the objects belonging to the framework domain, and how they interact, can help and shorten the analysis/modelling phase of possible new Parlay services; consequently the definition of new interfaces and improvements/enhancements of existing ones can take advantage of a FIM in terms of speed, easiness, completeness etc.

Vendors planning to develop Parlay Gateways have surely to face the problem of defining data/information models of their system, including a model of the framework functionality, but such models will be obviously tied to the specific implementation. In addition, parts of a FIM are already, implicitly, defined in the specification of current framework APIs, such as service subscription, service registration, and service properties.

The Framework Information Model, objective of the present work, to be useful in different contexts, should be carefully structured and defined at a proper detail level: not so deep to impose implementation constraints and not so abstract to hide aspects (namely entities, relationships, objects or other) that are needed to achieve the aforesaid objectives.

Scenarios:

This clause describes two possible scenarios/contexts that could take advantage of the FIM.

The Service Registration is a good example of a scenario that could take advantage in using a Framework model. Before a service can be accessed and used, it has to be registered in the Framework, where information on the available services is contained. The registration procedure is described in the specifications, as well as some relationships among the involved entities (namely Service Supplier Administrator, Framework, the Service that has to be registered); moreover some of the involved objects can be deduced from the interface definition, but a clear, high level view of the whole procedure is not available. Part of this issue could be covered by FIM.

The second scenario a FIM can be useful, is related to Service Subscription (this subject is treated in clause 4.17 of "Framework Interfaces, Client Application View – Version 2.1" specification). Even though the Service Subscription phase is described by the specification in some detail (e.g. in terms of a "Subscription Business Model", defining high level actors and relationships, with pictures and text), it is not simple to analyse all the relationships among the various involved entities (e.g. the Enterprise Operator, the Framework and the Client Application, that have precise roles in the Service Subscription phase), or to have a clear vision, as an example, of all the SAG (Subscription Assignment Group) aspects. A more formal and detailed representation, including the involved Framework entities that can be defined in the Framework model, can be helpful.

Requirements description:

A useful Framework Information Model should be able to support analysis (as mentioned before both standard-oriented and implementation oriented) of several aspects. In particular the following functionalities should be addressed:

1) Service Subscription and subscription management: objects and relationships description.

2) Service Interface and Framework Interface subscription by applications: each involved entity should be correctly defined, together with the configuration data related to application subscription. Possible aspects related SLA constraints to applications subscription should be addressed as well.

3) Services and service interfaces registration and configuration.

4) Service discovery.

5) Usage data management (e.g. logging data, Service Interface usage data).

Possible solution and further consideration:

Services and Service Interfaces configurations are typically done through properties. A property is composed of a name, a type and a value (data or policy). Moreover, each property can be tied to validation rules. The model should be able to cover these aspects (and consider their relationship with service properties).

A further need is that the model should be easily extensible, to allow Service Interfaces incremental introduction. Each Service Interface is characterized by set of properties. Some of such properties can be defined by standards, others can be proprietary, marking out a particular implementation.

The FIM definition will consist of a formal UML description, in terms of set of Class Diagrams describing objects and relationships; a textual description of the objects and their attributes will be given as well.

In addition to the model, other outputs can be proposals of enhancements to existing APIs as well as new APIs. As an example, a suggestion of API to manipulate some of the identified Framework objects, if useful, can be done.

5.2.2 Framework management tool

Source: Eurescom

Issue/Motivation:

The Information Model for Framework Functions should be accessible from some sort of management tool. To make this possible one should define the API to configure and access the data model.

Information from off-line Service Level Agreements and information needed for on-line Service Agreements should be entered via this API.