INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Series Q

Supplement 40

(11/2002)

SERIES Q: SWITCHING AND SIGNALLING

Technical Report: Reference document on API/object interface between network control and application layer

ITU-T Q-series Recommendations – Supplement 40

ITU-T Q-SERIES RECOMMENDATIONS

SWITCHING AND SIGNALLING

SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE	Q.1–Q.3
INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING	Q.4–Q.59
FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN	Q.60–Q.99
CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS	Q.100–Q.119
SPECIFICATIONS OF SIGNALLING SYSTEM No. 4	Q.120–Q.139
SPECIFICATIONS OF SIGNALLING SYSTEM No. 5	Q.140–Q.199
SPECIFICATIONS OF SIGNALLING SYSTEM No. 6	Q.250–Q.309
SPECIFICATIONS OF SIGNALLING SYSTEM R1	Q.310–Q.399
SPECIFICATIONS OF SIGNALLING SYSTEM R2	Q.400–Q.499
DIGITAL EXCHANGES	Q.500–Q.599
INTERWORKING OF SIGNALLING SYSTEMS	Q.600–Q.699
SPECIFICATIONS OF SIGNALLING SYSTEM No. 7	Q.700–Q.799
Q3 INTERFACE	Q.800–Q.849
DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1	Q.850–Q.999
PUBLIC LAND MOBILE NETWORK	Q.1000–Q.1099
INTERWORKING WITH SATELLITE MOBILE SYSTEMS	Q.1100–Q.1199
INTELLIGENT NETWORK	Q.1200–Q.1699
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR IMT-2000	Q.1700–Q.1799
SPECIFICATIONS OF SIGNALLING RELATED TO BEARER INDEPENDENT CALL CONTROL (BICC)	Q.1900–Q.1999
BROADBAND ISDN	Q.2000–Q.2999

For further details, please refer to the list of ITU-T Recommendations.

Supplement 40 to ITU-T Q-series Recommendations

Technical Report: Reference document on API/object interface between network control and application layer

Summary

There are many API/Object Interface-related activities outside of ITU-T SG11. Many API/Object Interface specifications have already been released from such activities and a lot of discussions for new APIs/Object Interfaces are starting within them. However, there is no good reference material to help introduce the API/Object Interface specifications under current discussion and it is difficult to know what kind of APIs are released or discussed. This Supplement provides high-level descriptions of API/Object Interface-related activities outside of ITU-T, covering the interface between network control and application layers. This Supplement should be used as a reference to the other API/Object Interface activities outside of ITU-T. This Supplement will also help to avoid the overlapping of the standardization effort.

Source

Supplement 40 to ITU-T Q-series Recommendations was prepared by ITU-T Study Group 11 (2001-2004) and approved under ITU-T Recommendation A.13 (10/2000) procedure on 22 November 2002.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this publication, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this publication may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the publication development process.

As of the date of approval of this publication, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this publication. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Page

1 Scope 1

2 References 2

2.1 Website References 2

2.2 Document References 2

3 Definitions 2

4 Abbreviations 2

5 Activity in each standard body 2

5.1 Parlay/ETSI/3GPP 2

5.2 JAIN 9

5.3 OMG 13

5.4 TINA 18

6 Applicability 21

6.1 Objective 21

6.2 Classification of API 22

6.3 Applicability 22

Appendix I 25

I.1 Parlay/ETSI/3GPP 25

I.2 JAIN 26

I.3 OMG 26

I.4 TINA 27

Supplement 40 to ITU-T Q-series Recommendations

Technical Report: Reference document on API/object interface between network control and application layer

1 Scope

This Supplement provides high level descriptions of API/Object Interface related activities outside of ITU-T and clarifies the applicability of each API/Object Interface specification. It focuses on the specifications of API/Object Interface between network control and application layers. It will also help to avoid the overlapping of the standardization effort. The scope of this Supplement is shown in Figure 1.

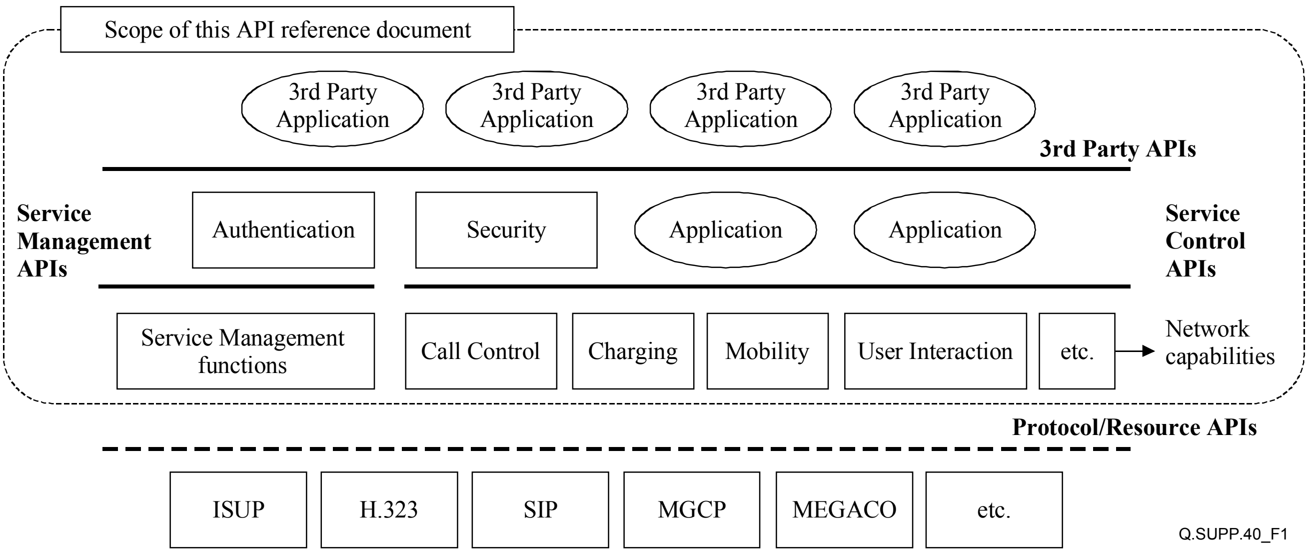

Figure 1 – Reference layered model of network

The outline of Each API category is as follows.

3rd Party APIs: The APIs in this category are the APIs open for 3rd Party and they provide authentication and security functions. 3rd Party can utilize Service Control APIs and Service Management APIs through these APIs.

Service Control APIs: The APIs in this category provide network capabilities to control services. The example APIs included in this category are Call Control, Charging, Mobility, User Interaction, etc. Network operators or 3rd Party can develop their applications making use of these APIs.

Service Management APIs: The APIs in this category provide service management functions such as to support execution, deployment and observation of services, etc. The example APIs included in this category are Service Execution, Deployment, etc. Network operators can manage services based on these APIs.

Protocol/Resource APIs: The APIs in this category provide protocol/resource oriented functions to network capability functions. The APIs in this category are not within the scope of this Supplement.

2 References

2.1 Website References

[w1] Parlay website: http://www.parlay.org/

[w2] ETSI website: http://docbox.etsi.org/span/open/span12/osa.html

[w3] 3GPP website: http://www.3gpp.org/

[w4] JAIN website: http://java.sun.com/products/jain/

[w5] OMG website: http://www.omg.org/

[w6] TINA website: http://www.tinac.com

2.2 Document References

[d1] 3GPP TS 21.903 Vocabulary document

[d2] Parlay website for downloading documents: http://www.parlay.org/specs/index.asp

[d3] ETSI website for downloading documents: http://pda.etsi.org/pda/queryform.asp

[d4] 3GPP website for downloading documents: http://www.3gpp.org/TB/cn/cn5/specs.htm

[d5] JAIN website for downloading documents: http://java.sun.com/products/jain/api_specs.html

[d6] JAIN White Paper: http://java.sun.com/products/jain/wp_articles.html

[d7] OMG website for downloading documents: http://www.omg.org/technology/documents/spec_catalog.htm

[d8] TINA website for downloading documents: http://www.tinac.com/specifications/specifications.htm

3 Definitions

The definitions in this Supplement follow the definitions in each standard body (e.g. 3GPP vocabulary document [d1]).

4 Abbreviations

The abbreviations in this Supplement follow the abbreviations in each standard body (e.g. 3GPP vocabulary document [d1]).

5 Activity in each standard body

This clause provides introduction to each activity of API-related standard bodies.

5.1 Parlay/ETSI/3GPP

5.1.1 Overview of joint activity

5.1.1.1 Overview

The Parlay Group [w1], ETSI [w2], and 3rd Generation Partnership Project (3GPP) [w3] are working jointly to define common API for Open Service Access (OSA). Hereafter, the API developed by this joint group is called OSA/Parlay API.

A detailed description of each standard body is given in the following clauses.

The OSA/Parlay APIs are intended to enable a new generation of off-the-shelf network applications and components (e.g. messaging, mobility, end-to-end quality of service) to be developed by application providers (Independent Software Vendor (ISV)/Application Service Provider (ASP)) independent of the underlying voice/multimedia network. Faster time-to-market and a less complex development cycle are some of the expected key benefits of the OSA/Parlay APIs. The OSA/Parlay APIs consist of two categories of interface:

- Service Interfaces. These offer applications access to a range of network capabilities and information.
- Framework Interfaces. These provide the supporting capabilities necessary for the Service Interfaces to be secure and manageable.

5.1.1.2 Description

The OSA/Parlay API uses the Unified Modelling Language (UML) to describe access to Third Party Service applications.

The API is not a piece of code but provides a mechanism whereby objects transparently make requests to, and receive responses from, other objects on different platforms in heterogeneous environments like Intelligent Networks.

Where the OSA/Parlay API is located in the network is shown in Figure 2.

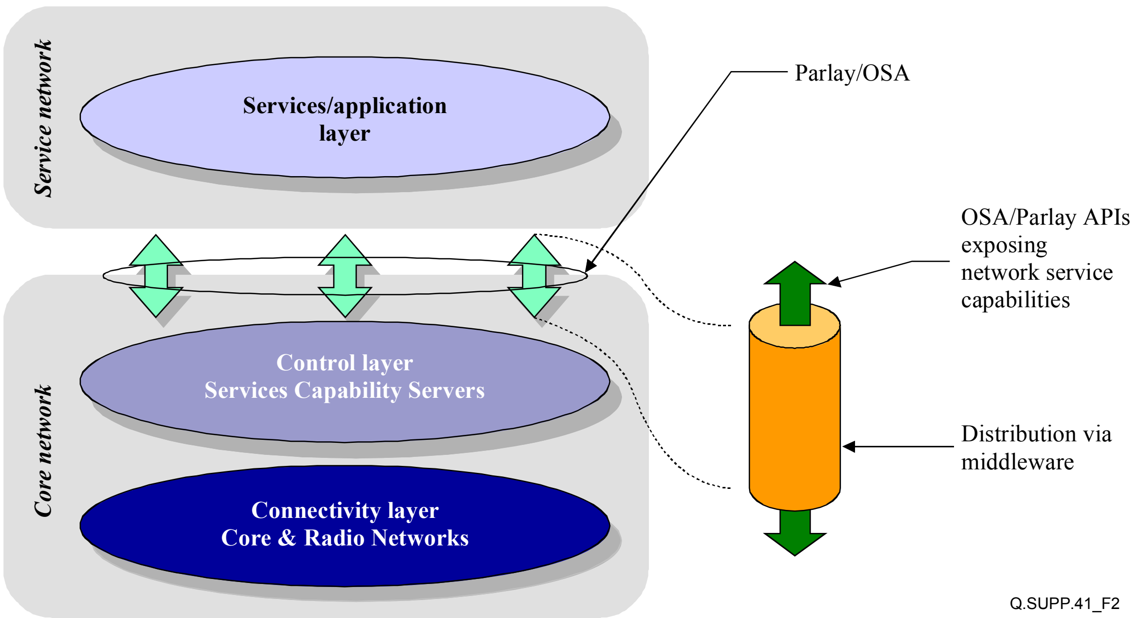

Figure 2 – Where the Parlay/OSA API is located

The architecture of Parlay/OSA API is shown in Figure 3.

Functions provided by the service interfaces allow access to traditional network capabilities such as call management, messaging, and user interaction. The service interfaces also include generic application interfaces to ease the deployment of communications applications.

Functions provided via the framework interfaces in the Parlay/OSA API specifications:

- Service Registration & subscription & discovery;
- Authentication and Authorisation;
- Integrity Management.

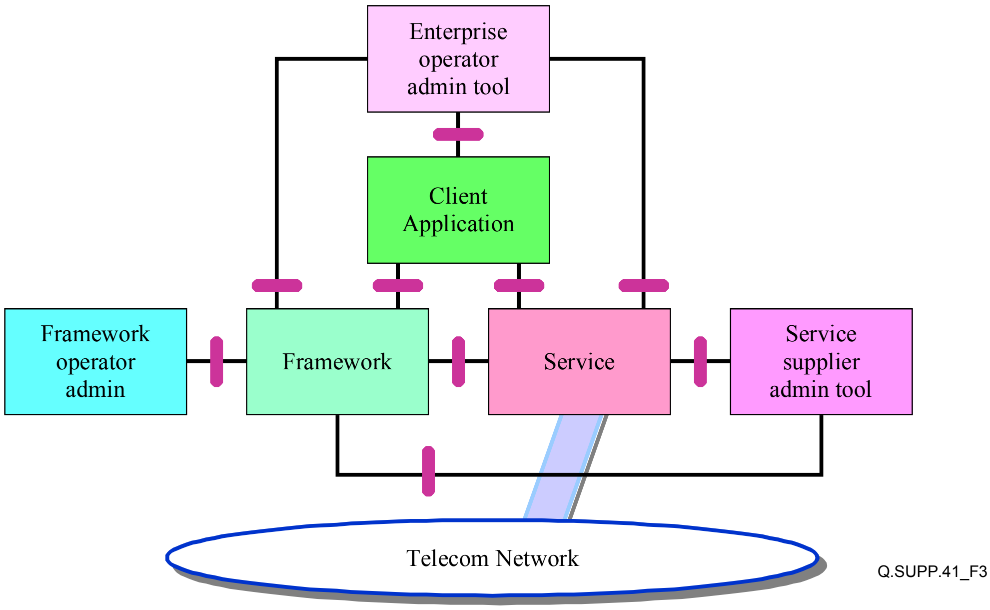

Figure 3 – The architecture of the Parlay/OSA APIs

The current Parlay/OSA API Service Capability Features (SCFs) are shown in Table 1.

Table 1 – Current Parlay/OSA SCFs

Call Control	The Call control family, with capabilities ranging from setting up basic calls to manipulating multimedia conference calls
Framework	Infrastructure capabilities such as authentication, SCF discovery, SCF registration, fault management, etc.
User Interaction	Obtain information from the end-user, play announcements, send short text messages, etc.
User location/User status	Obtain location and status information
Terminal capabilities	Obtain the capabilities of an end-user terminal
Data session control	Control of data sessions
Generic Messaging	Access to mailboxes
Connectivity Management	Provisioned QoS
Account Management	Access end-user accounts
Content based Charging	Charge end-users for use of applications / data
Policy Management	Tasks include the creation, update, deletion and viewing of Policy information
Presence and Availability Management	Management of information relating to presence, such as the dynamic state of devices / software and their owners

The API is object-oriented and consists of several categories of interfaces.

Generic service interfaces

The API is split into two types of interface class descriptions, Service and Framework. Framework classes are those that are perceived to be applicable to the interface irrespective of the type of service that is being implemented at a specific moment in time e.g. Authentication, whereas Service Interface classes are individual services that may be required by the client or network operator to enable the running of third party applications over the interface, e.g. Messaging type service.

Each of the service capability feature description defines the interfaces, parameters and state models that form part of the API specification. UML is used to specify the interface classes. As such, it provides a UML interface class description of the methods (API calls) supported by that interface and the relevant parameters and types.

Framework interfaces

The Framework is split into two different sections, the first addressing the Client view. The second addresses the relationship between the Service and Framework providers. The client-to-Framework section is split into 5 parts, these being; Trust and Security Framework (which includes Authentication), Fault Management, Integrity Management, Service Subscription and Service Discovery. The Service-to-framework interface contains all of the same interfaces except for Service Subscription.

Service data definitions

Service Data Definitions provides the Data Definitions necessary to support the Generic Service interface. For instance the Generic Call Control Service Data Definitions describes each of the Data types that were shown in the detailed parameter descriptions made in the 'Generic Call Control Service Interface' part and so on.

Framework data definitions

Framework Data Definitions once again provide the Data Definitions necessary to support the Framework interface.

Common data definitions

Common Data Definitions provide the Data definitions that are common to both the Framework and Generic Service API parameters.

Sequence Transition Diagrams (STDs)

Sequence Transition Diagrams contains the sequence transition diagrams from each service. They are used to enhance the understanding of each service in more detail.

OMG IDL

OMG IDL provides an OMG IDL version of the whole API. It was felt useful that a working version of the API be produced so that the API could be realizable in the marketplace of today.

5.1.2 Parlay

5.1.2.1 Overview

The Parlay Group is an open, multi-vendor forum organized to create open, technology-independent Application Programming Interfaces (APIs) which enable IT companies, ASPs, ISVs, Internet Companies, e-Business Companies, software creators, service bureaux, and large and small enterprises as well as network providers, network equipment vendors and application suppliers to develop applications across multiple networks.

The Parlay Group is studying the enhancement of its APIs for enlarging their field of application in addition to Parlay/OSA APIs discussed in the joint group.

5.1.2.2 Description

The architecture of Parlay API is the same as that of Parlay/OSA API and it is shown in Figure 3.

The implementation of Parlay is based on application servers outside the network domain, running Parlay applications. A Parlay Gateway, provided by the network operator, ensures secure, manageable access to capabilities in the service provider’s network (see Figure 4).

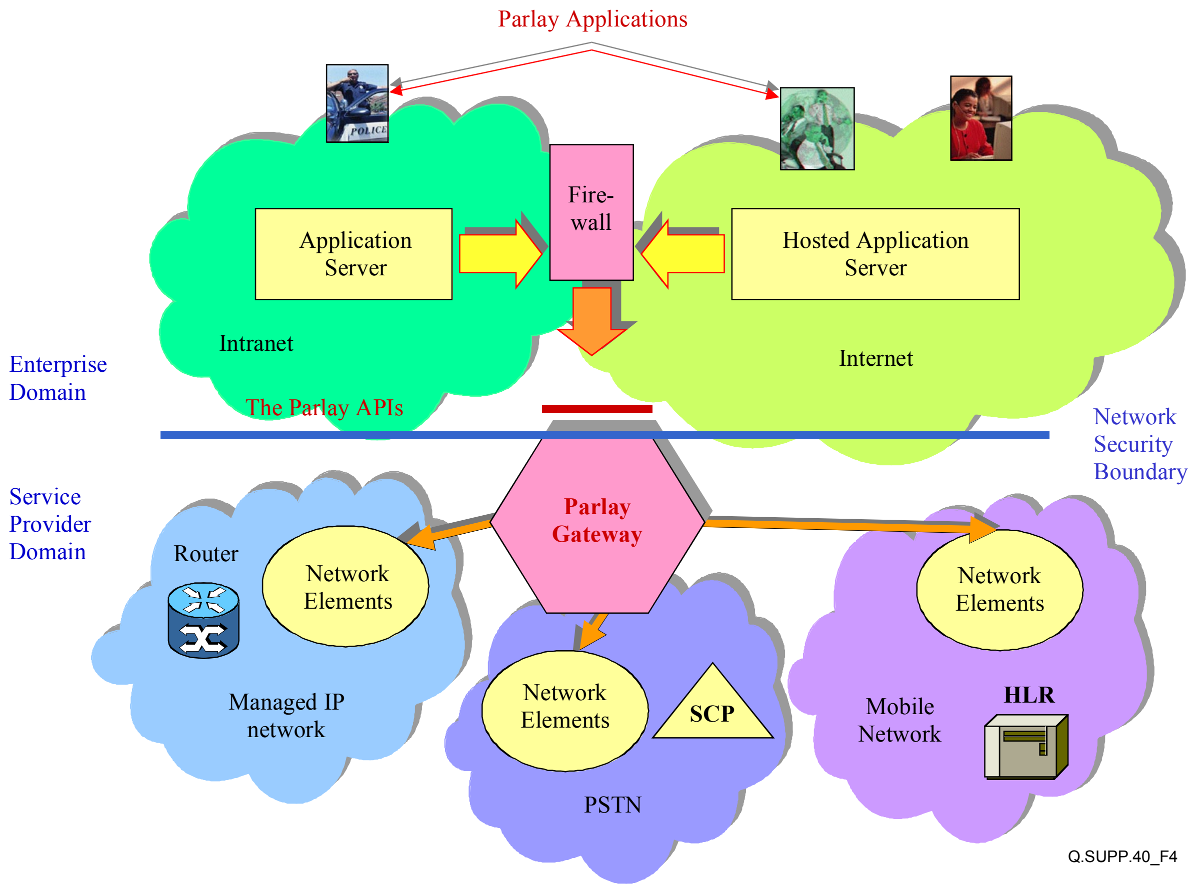

Figure 4 – Implementing Parlay in the network

The content of the APIs covered by Parlay is on the Parlay website [w1].

5.1.2.3 Released specification

Released specifications of Parlay APIs can be seen at Parlay website for available documents [d2]. The latest released specifications are listed in an Appendix I.

5.1.2.4 Schedule

Schedule of Parlay API Specifications can be seen at Parlay website [w1]. The latest schedule is listed in an Appendix I.

5.1.2.5 Relationship with other bodies

– relationship with ETSI and 3GPP is described in 5.1.1;

– cooperation with JAIN Community member companies on SPA;

– cooperation with OMG on Framework.

5.1.3 ETSI

5.1.3.1 Overview

ETSI (European Telecommunication Standards Institute) is a not-for-profit organization whose mission is to produce the telecommunication standards that will be used for decades to come throughout Europe and beyond. Based in Sophia Antipolis, south of France, ETSI unites 874 members from 54 countries inside and outside Europe, and represents administrations, network operators, manufacturers, service providers, research bodies and users.

Among its technical committees, this area of work is supported by: Services and Protocols for Advanced Networks (SPAN). This Committee was formed by the merger of Signalling Protocols and Switching (SPS) and Network Aspects (NA) in April 1999. As its name suggests, ETSI SPAN covers the areas formerly the responsibility of NA; in particular, the definition of services offered by telecommunication networks of all types and of SPS; in particular, signalling protocols for ISDN.

Parlay/OSA API discussed in the joint group is specified as a standard of ETSI.

5.1.3.2 Description

The architecture of OSA API specified as ETSI Standard is the same as that of Parlay/OSA API and it is shown in Figure 3.

5.1.3.3 Released specification

Released API specifications developed by ETSI can be seen at ETSI website for available documents [d3]. The latest released specifications are listed in an Appendix I.

5.1.3.4 Schedule

Schedule of API specification work at ETSI can be seen at ETSI website [w2]. The latest schedule is listed in an Appendix I.

5.1.3.5 Relationship with other bodies

– relationship with Parlay and 3GPP is described in 5.1.1;

– cooperation with JAIN Community member companies on SPA.

5.1.4 3GPP

5.1.4.1 Overview

The Partners have agreed to cooperate in the production of globally applicable Technical Specifications and Technical Reports for a 3rd Generation Mobile System based on evolved GSM core networks and the radio access technologies that they support (i.e., Universal Terrestrial Radio Access (UTRA) both Frequency Division Duplex (FDD) and Time Division Duplex (TDD) modes).

The Partners have further agreed to cooperate in the maintenance and development of the Global System for Mobile communication (GSM) Technical Specifications and Technical Reports including evolved radio access technologies (e.g. General Packet Radio Service (GPRS) and Enhanced Data rates for GSM Evolution (EDGE)). The Project is called the "Third Generation Partnership Project" and may be known by the acronym "3GPP".

As a part of whole 3GPP's specifications, it has specified the Open Service Access providing open service APIs.

3GPP OSA API is a part of OSA/Parlay API, which is focused on the object of 3GPP.

5.1.4.2 Description

The Open Service Access (OSA) defines an architecture that enables operator and third party applications to make use of network functionality through an open standardized API (the OSA API) [d4]. Its architecture is the same as Parlay/OSA API and it is shown in Figure 3. Figure 5 shows an overview of the Open Service Access being specified in 3GPP.

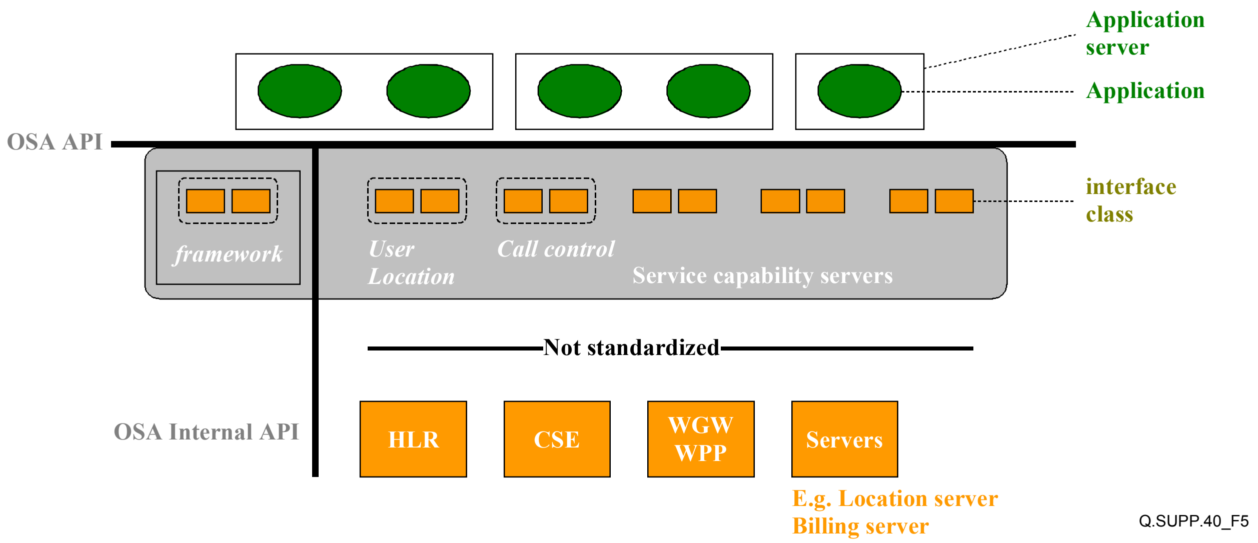

Figure 5 – Architecture of open service access

The Open Service Access consists of three parts:

– **Applications**: typical applications are VPN, conferencing and location based applications. These applications are implemented in one or more Application Servers;

– **Framework**: providing applications with basic mechanisms that enable them to make use of the service capabilities in the network. Examples of framework service capability features are Authentication and Discovery. Before an application can use the network functionality made available through Service Capability Features, authentication between the application and framework is needed. After authentication, the discovery service capability feature enables the application to find out which network service capability features are provided by the Service Capability Servers. The network service capability features are accessed by the methods defined in the OSA interfaces;

– **Service Capability Servers**: providing the applications with service capability features, which are abstractions from underlying network functionality. Examples of service capability features offered by the Service Capability Servers are Call Control and User Location. Similar service capability features may possibly be provided by more than one Service Capability Server. For example, Call Control functionality might be provided by SCSs on top of CAMEL and MExE.

The OSA service capability features are specified in terms of a number of interfaces and their methods. The interfaces are divided into two groups:

– framework interfaces;

– network interfaces.

5.1.4.3 Released specification

3GPP's specifications are organized into series according to their function. Under the responsibility of 3GPP are the GSM specifications transferred from ETSI Technical Committee SMG; and the third generation specifications originated in 3GPP.

The specifications are batched in "Releases". Each new Release adds further functionality to enhance the capabilities of networks and terminals built to that Release's specifications. Using all the Technical Specifications (and Technical Reports) for a given Release, systems may be built to conform to the requirements of that Release.

Released API specifications developed by 3GPP can be seen at 3GPP website for available documents [d4]. The latest released specifications are listed in Appendix I.

5.1.4.4 Schedule

New features included in each new Release are defined in the project plan.

The project plan and all Change Requests agreed by Working Groups are approved at plenary TSG meetings, and the resulting specifications are made available after each TSG SA meeting, four times a year. The status of the specifications at these points in time is given in the corresponding "status list".

Schedule of API specification work at 3GPP can be seen at 3GPP website [w3]. The latest schedule is listed in an Appendix I.

5.1.4.5 Relationship with other bodies

– relationship with Parlay and ETSI is described in 5.1.1;

– cooperation with JAIN Community member companies on SPA.

5.2 JAIN

5.2.1 Overview

The JAIN APIs are a set of Java technology-based APIs which enable the rapid development of Next Generation telecom products and services on the Java platform [w4]. The JAIN APIs bring service portability, convergence, and secure network access to telephony and data networks.

By providing a new level of abstraction and associated Java interfaces for service creation across Public Switched Telephone Network (PSTN), packet (e.g. Internet Protocol (IP) or Asynchronous Transfer Mode (ATM)) and wireless networks, JAIN technology enables the integration of Internet (IP) and Intelligent Network (IN) protocols. Furthermore, by allowing Java applications to have secure access to resources inside the network, the opportunity is created to deliver thousands of services rather than the dozens currently available. Thus, JAIN technology is changing the telecommunications market from many proprietary closed systems to a single network architecture where services can be rapidly created and deployed.

JAIN technology is being specified as a telecom oriented to the Java Platform. Development is being carried out under the terms of Sun's Java Specification Participation Agreement (JSPA), Java Community Process^{SM} (JCP), and Sun's Community Source Code Licensing (SCSL) terms. For further information on the JCP please visit http://java.sun.com/aboutJava/communityprocess/.

The JAIN initiative consists of two API Specification areas of development:

- The Protocol API Specifications specify interfaces to wireline, wireless and IP signalling protocols;
- The Application API Specifications address the APIs required for service creation within a Java framework spanning across all protocols covered by the Protocol API Specifications.

The objective of the JAIN initiative is to create an open value chain from 3rd-party service providers, facility-based service providers, telecom providers, and network equipment providers to telecom, consumer and computer equipment manufacturers.

The JAIN initiative integrates wireline, wireless, and packet based networks, as illustrated in Figure 6. The adaptation of network specific protocols to the JAIN model is covered in the Protocol API Specifications. Additionally, the JAIN initiative abstracts the protocols covered by the Protocol API Specifications into a single call control, coordination, and transaction model to be used by compliant services. This is being driven by the work in the Application API Specifications.

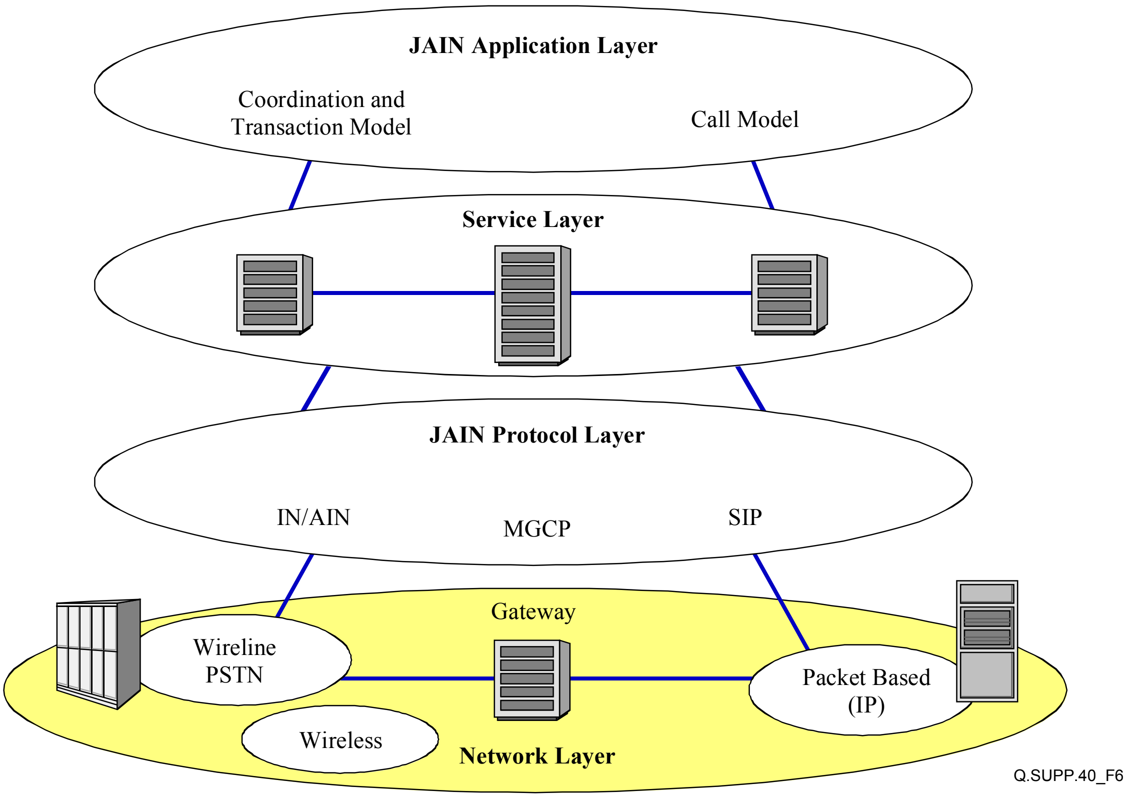

Figure 6 – JAIN Initiative

The JAIN related information in this clause was captured from JAIN White paper [d6].

Sun, Sun Microsystems, Java, JAIN, JINI, JavaBeans, and Java Community Process are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries.

5.2.2 Description

5.2.2.1 Architecture

The JAIN Architecture is shown in Figure 7.

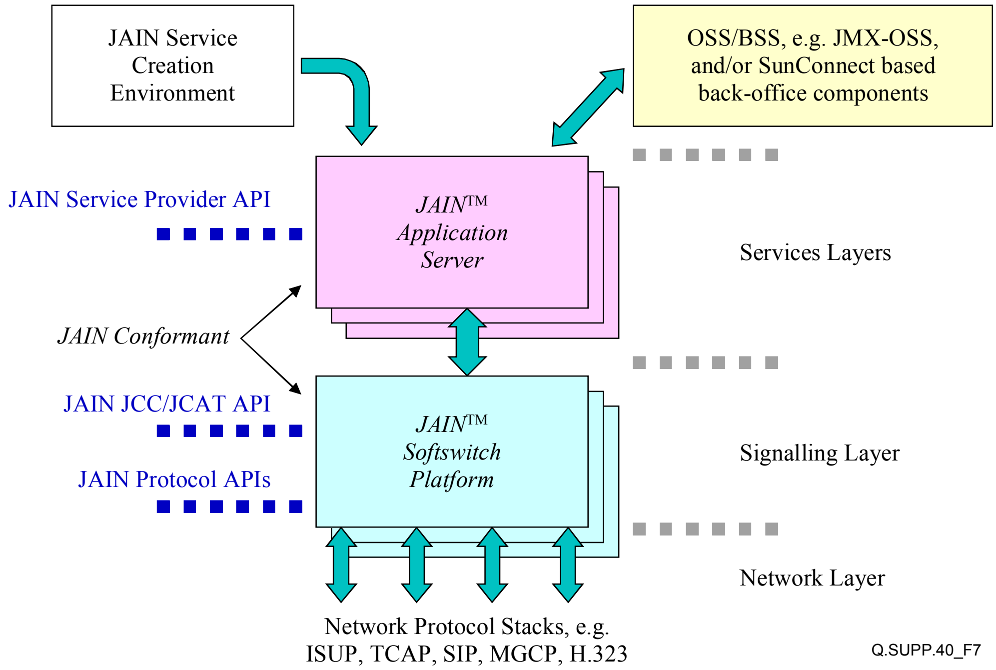

Figure 7 – JAIN Architecture

NOTE – JAIN Protocol APIs are outside the scope of this Supplement.

At its core, the JAIN architecture defines a software component library, a set of development tools, a service creation environment, and a carrier-grade service logic execution environment to build next generation services for integrated PSTN, packet (e.g. ATM and IP), and wireless networks.

As illustrated in Figure 7, the JAIN architecture includes a Service Creation Environment (SCE) for both trusted next generation network services and untrusted third-party applications. Trusted services (and policies) reside within the core of public networks. Untrusted services are services written by third parties that access functions within the core public networks. Through a secure service provider interface, these third party applications are kept from compromising the reliability or integrity of those networks.

The JAIN architecture provides for the (trusted) next generation network services in a carrier-grade service logic execution environment It is expected that many services and implementations of the JAINSLEE will be implemented using Enterprise JavaBeans™ (EJB).

Untrusted services like trusted services might have similar requirements for the SLEE depending on their scope. Untrusted services also will rely on container infrastructures that can be used to 'hold' services (e.g. EJB, Java™ Embedded Server (JES), and JINI™). The JAIN SCE is compatible with both environments (e.g. trusted and untrusted service creation).

Figure 8 is a pictorial representation of where JAIN APIs are defined within a communications platform. The softswitch architecture is centered on mapping the call control/session interfaces onto the underlying protocol APIs. Since softswitches perform signalling on IP networks, most are equipped with a SIP, MGCP, MEGACO or H.323 underlying protocols. Several softswitches also include SS7 protocols to address interfaces for the existing telephone network.

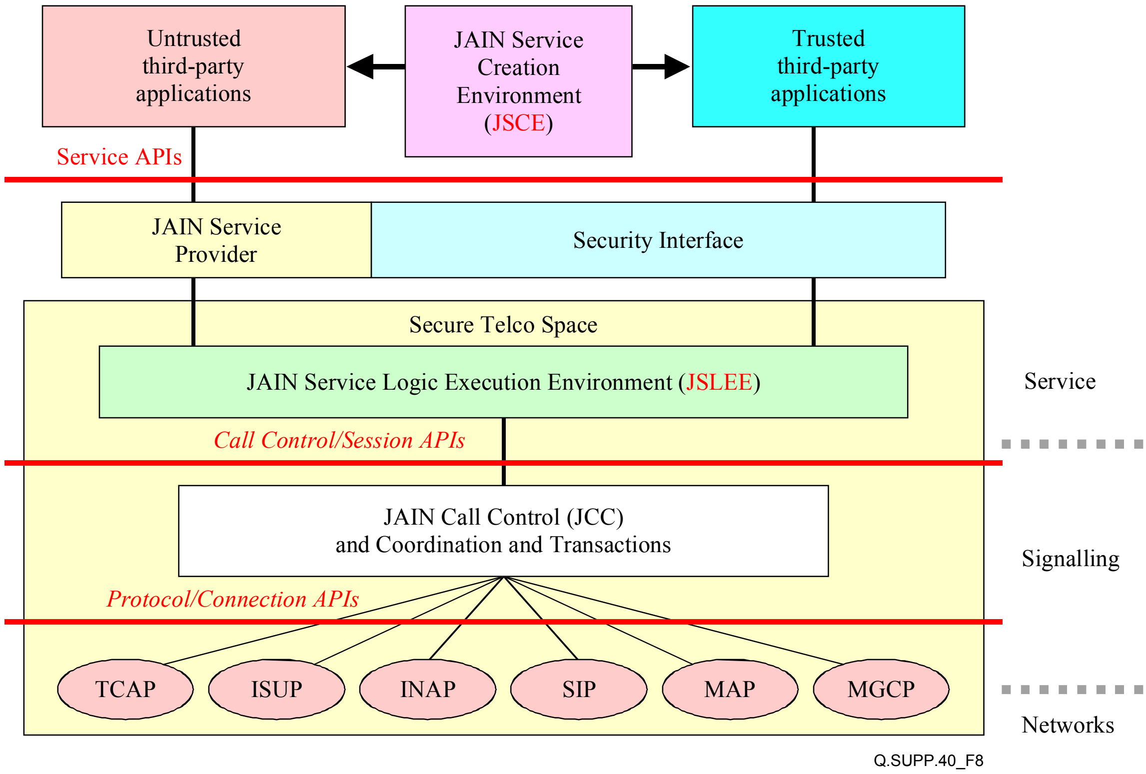

Figure 8 – JAIN APIs

5.2.2.2 JAIN API specifications

JAIN™ Call Control

The JAIN Call Control (JCC) provides applications with a consistent mechanism for interfacing with underlying divergent networks. The application needs only interface once to a JCC interface and the subsequent JAIN adapters will allow calls and data to pass to various networks.

JAIN™ Coordination and Transaction

Java™ Coordination and Transaction (JCAT) includes (but is not limited to) the facilities required for applications to be invoked and return results before, during or after calls; to process call parameters or subscriber-supplied information; and to engage in further call processing and control.

JCAT perceives Java™ Call Control (JCC) as its core call control model. JCAT extends the JCC call control model with terminal capabilities and it enriches JCC's state diagrams such that an even more diverse range of applications can be supported. The AIN/IN class of applications is an example of applications that can be supported.

JAIN™ Service Provider API for the Parlay Specification

The JAIN Service Provider APIs (SPA) for the Parlay specification will provide the secure access mechanism to network capabilities. This set of APIs will focus on a Java technology based implementation of Parlay and with extensibility to allow other services to be exported by the network operator and discovered by the service provider/user.

JAIN SPA APIs will be defined using the set of rules provided by the Java Realization work group Parlay.

JAIN™ Service Logic Execution Environment

Once services are created, they can be tested and deployed in the JAIN Service Logic Execution Environment (SLEE). JAIN SLEE defines interfaces and requirements mandatory for telco/Internet operations within carrier grade and Internet networks.

JAIN™ Service Creation Environment

This Java™ Specification Request (JSR) defines the standard software interfaces of the Service Creation Environment (SCE) for JAIN™. The JAIN™ SCE is a set of software interfaces to support and simplify the creation of portable telecommunication services delivered primarily to the JAIN™ Service Logic Execution Environment (JAIN™ SLEE), while not restricted to this class of SLEEs.

JAIN™ Common API

Currently there are multiple different JAIN JSRs which are in the Java Community Process. As a result of tight collaboration among the members of the JAIN initiative, these JSRs have a consistent architecture and they share common design patterns. As a result, they have common base interfaces and classes, such as data types and exception definitions.

This specification will document and specify these common base interfaces and classes to avoid duplication of these interfaces and classes in each of the above JAIN JSRs and to maintain consistency of these interfaces and classes across these JSRs.

The JAIN APIs also include Protocol APIs, which are outside the scope of this Supplement.

5.2.3 Released specification

Released specifications of JAIN APIs can be seen at JAIN website for available documents [d5]. The latest released specifications are listed in an Appendix I.

5.2.4 Schedule

Schedule of JAIN API Specifications can be seen at JAIN website [w4]. The latest schedule is listed in an Appendix I.

5.2.5 Relationship with other bodies

– coordination with Parlay, ETSI and 3GPP on SPA.

5.3 OMG

5.3.1 Overview

The OMG was formed to create a component-based software marketplace by hastening the introduction of standardized object software [w5]. The organization's charter includes the establishment of industry guidelines and detailed object management specifications to provide a common framework for application development. Conformance to these specifications will make it possible to develop a heterogeneous computing environment across all major hardware platforms and operating systems. These specifications are used worldwide to develop and deploy distributed applications for vertical markets, including Manufacturing, Finance, Telecoms, Electronic Commerce, Real-time systems and Health Care.

The Object Management Group (OMG) is an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications. The OMG is moving forward in establishing Common Object Request Broker Architecture (CORBA) as the "Middleware that's Everywhere" through its worldwide standard specifications: CORBA/IIOP, Object Services, Internet Facilities and Domain Interface specifications, UML and other specifications supporting Analysis and Design.

5.3.2 Description

5.3.2.1 Object management architecture

OMG's standardization is based on the Object Management Architecture shown in Figure 9 [d7]. The Object Management Architecture (OMA) is a set of standard interfaces for standard objects that support CORBA applications. It includes the base-level CORBA services, the CORBA facilities, and a large and growing set of Domain Specifications. To cover the wide variety of its execution environment, the Minimum CORBA (a static subset of CORBA targeting primarily an embedded system), the Real-Time CORBA and Fault-Tolerant are also adopted by OMG.

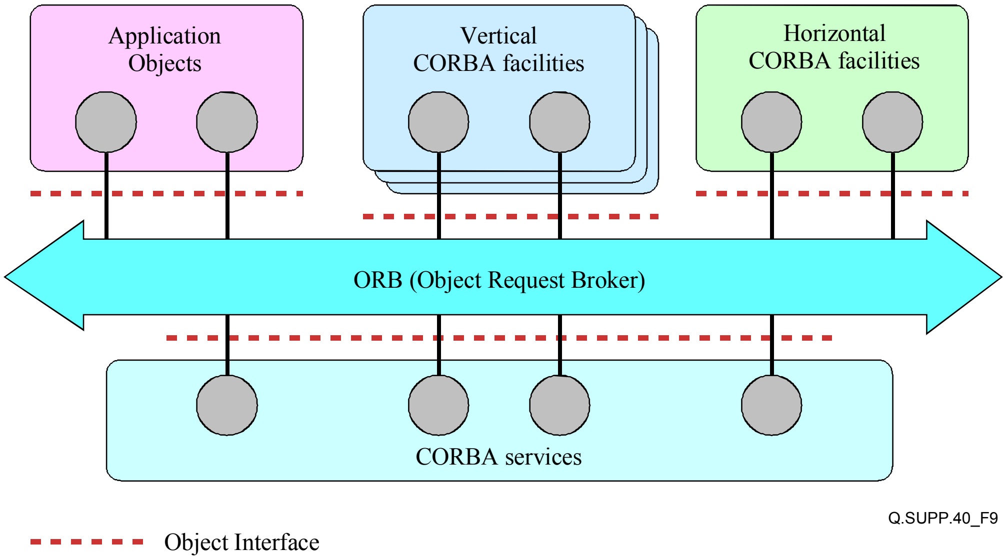

Figure 9 – OMA: Object management architecture

5.3.2.2 Unified modelling language

The Unified Modelling Language (UML) standardizes representation of system analysis and design. The UML covers a set of Structural Diagrams (Class, Component, Object, and Deployment Diagrams), a set of Behavioral Diagrams (Use Case, Statechart, Sequence, Activity, and Collaboration Diagrams) and a set of Model Management Diagrams (Packages, Models, Subsystems).

UML is supported by the Meta-Object Facility (the MOF), and XML Metadata Interchange (XMI), which standardize interchange of a target model from one design tool to another, and to other tools as related development progresses.

5.3.2.3 Inter-ORB communication

Protocol-level ORB Interoperability is supported by General Inter-ORB Protocol (GIOP) and Internet Inter-ORB Protocol (IIOP). While GIOP specifies protocol-independent Inter-ORB communication, IIOP specifies realization of GIOP over TCP/IP including management of its connection. Relationship among ORB, GIOP and IIOP is illustrated in Figure 10.

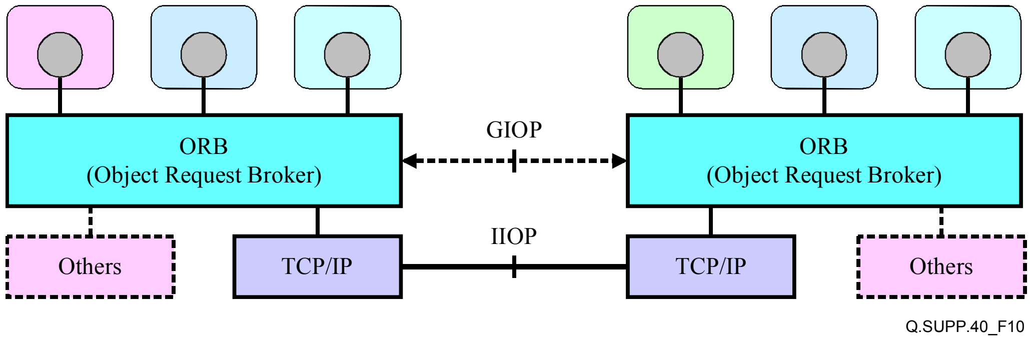

Figure 10 – GIOP and IIOP

5.3.2.4 CORBA component model

CORBA Component Model (CCM) provides a secure and manageable framework for server applications. Key concept of CCM is the CORBA Container illustrated in Figure 11, and it provides common framework for managing CORBA Components/Enterprise Java Beans (EJB) as well as for accessing CORBA Services, i.e. transactions, security, events, and persistence.

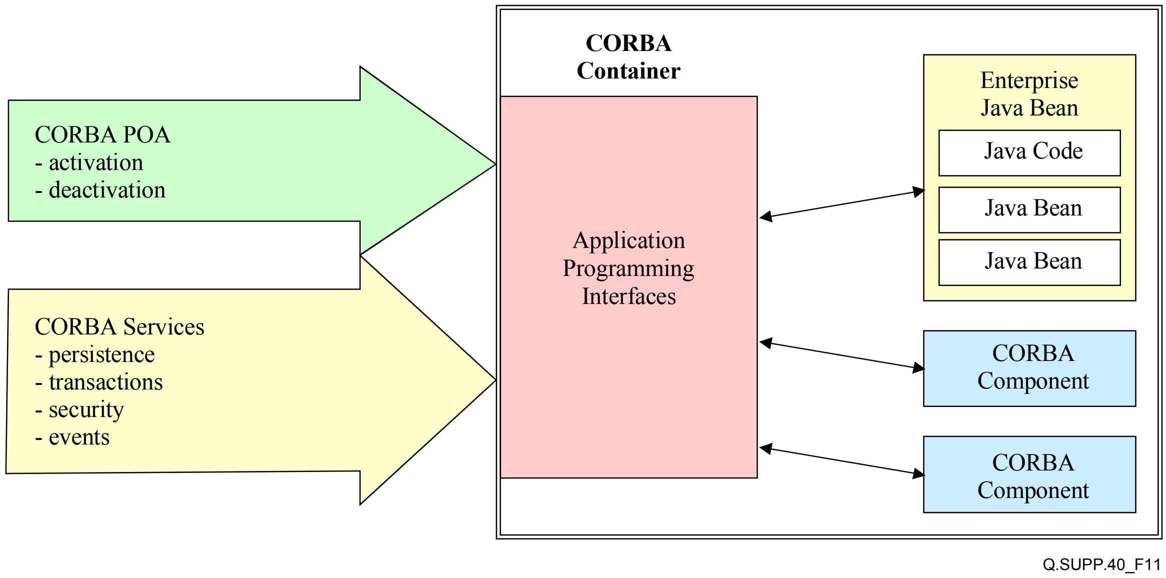

Figure 11 – CORBA container and CORBA components

5.3.2.5 Model driven architecture

The latest movement in OMG standardization is the development of "MDA: Model Driven Architecture" as illustrated in Figure 12. As illustrated in the figure, MDA is based on the following four major OMG standards:

– UML: Unified Modelling Language (see 5.3.2.2)

– MOF: Meta-Object Facilities: providing an abstract model called a meta-metamodel

– XMI: XML Metadata Interchange: specifying an open information interchange model using XML syntax

– CWM: Common Warehouse Model: establishing a common metamodel for warehousing and standardizing the syntax and semantics needed for import, export, and other dynamic data warehousing operations.

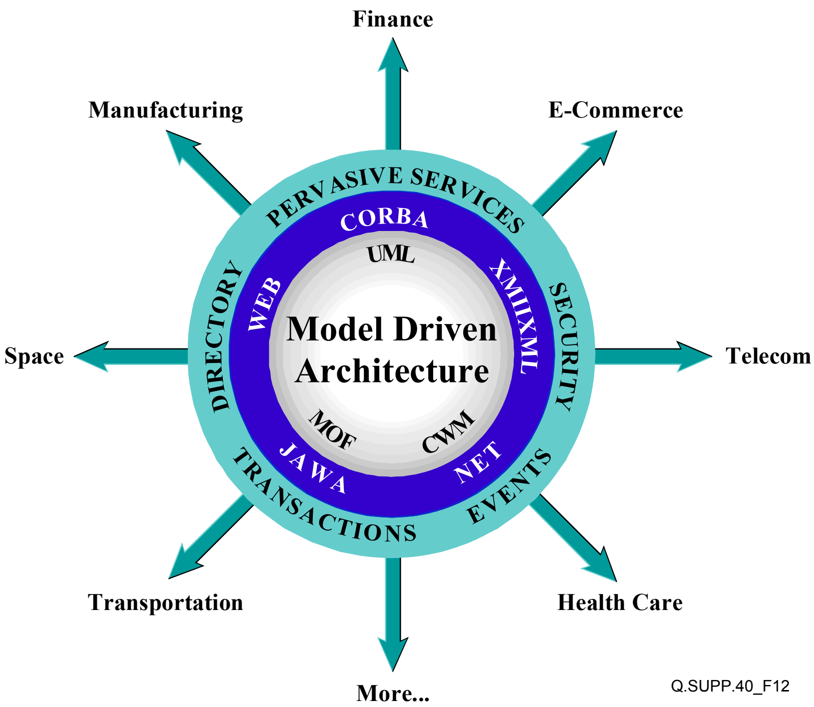

Figure 12 – Overview of Model Driven Architecture (MDA)

In MDA, these technologies enable developing platform independent model of target systems. The platform independent model supported by the MDA can be easily mapped into any platform-dependent model, e.g. CORBA model, Java/EJB model, XML/SOAP model for the development of real implementation based on the selected platform.

5.3.2.6 Application-wise specifications

Relevant Application-wise specifications identified are the following:

1) *Telecom Service Access and Subscription facility (TSAS) [d7], OMG Domain Specifications*

In this specification, three domains, Consumer, Retailer and Service Provider, are defined. For the Consumer domain, it assumes two roles, a subscriber and an end-user. TSAS offers mechanisms to establish and release authenticated connections between different domains in terms of "User" and "Provider" relationship. Therefore, TSAS is applicable to both interactions between Consumers and Retailer Domains and interactions between Retailer and Service Provider Domains. Figure 13 illustrates the basic model used in TSAS.

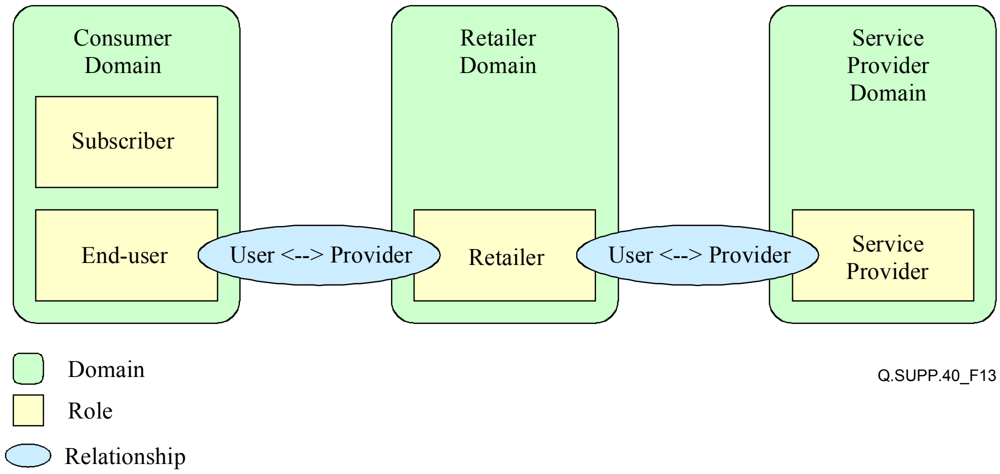

Figure 13 – TSAS domains, roles and relationship

The usage of services can be structured in two sessions, an Access session and a Service session, and TSAS specification distinguishes these two different sessions.

2) *CORBA/TC Interworking and SCCP-Inter ORB Protocol [d7], OMG Domain Specifications*

This specification addresses the interworking between CORBA-based Intelligent Network (IN) applications and the existing IN infrastructure. Although the principal application of this specification is for IN applications, its solution is of general applicability to any TC User, e.g. Mobile Application Part (MAP). Two scenarios for the use of CORBA in IN signalling are:

i) TC/CORBA Application Interworking;

ii) GIOP Mapping onto connectionless SCCP.

Figure 14 illustrates an example of the configuration of scenario i) "TC/CORBA Application Interworking". Here, the interworking of CORBA-based IN Application Entities (e.g. Service Control Point (SCP)) with legacy IN Application Entities (e.g. Service Switching Point (SSP)) is supported through a gateway mechanism that provide a CORBA view of a legacy target and a legacy view of a CORBA target.

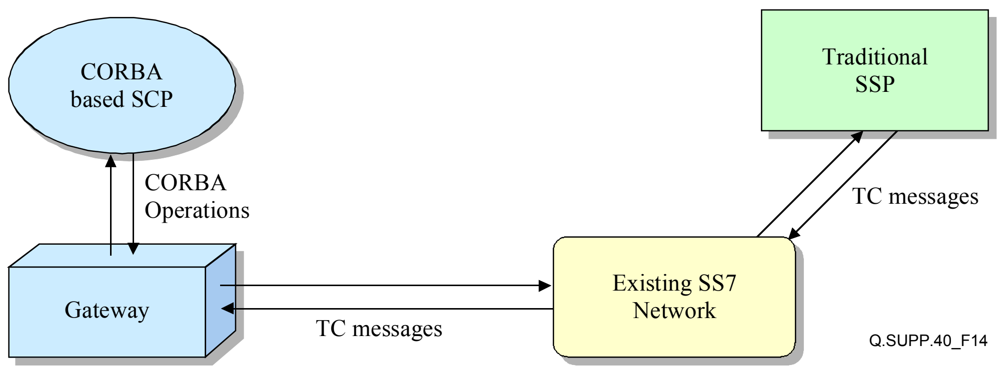

Figure 14 – Interworking between CORBA-based IN application and traditional IN applications (Scenario i)

Figure 15 illustrates an example of the configuration of scenario ii) "GIOP Mapping onto connectionless SCCP". Here, the interworking of CORBA-based IN Application Entities with using existing Signalling System 7 (SS7) infrastructure as a transport network for GIOP messages is supported. For this scenario, a Specification for mapping GIOP messages onto connectionless SCCP in the SS7 stack is defined.

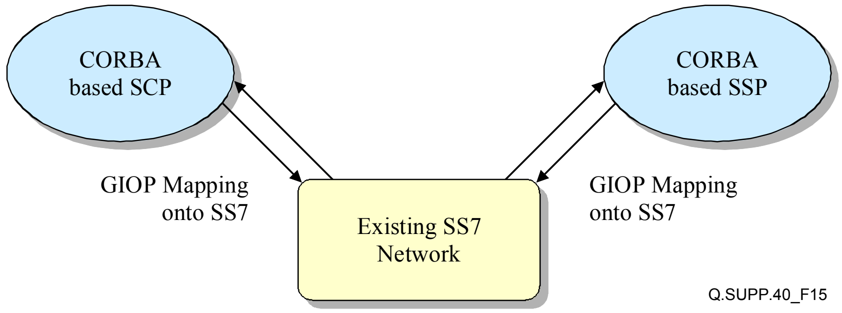

Figure 15 – Interworking between islands of CORBA-based IN applications using the SS7 network (Scenario ii)

5.3.3 Released specification

Overview of OMG specifications is given by the following URL:

http://www.omg.org/technology/documents/spec_catalog.htm

5.3.4 Schedule

The following URL shows the latest schedule on its specifications:

http://www.omg.org/news/meetings/tc/Tech_Adoption/index.htm

5.3.5 Relationship with other bodies

OMG has close collaboration with many standard bodies including Parlay (Framework), TeleManagement Forum, Foundation for Intelligent Physical Agents, ISO JTC1/SC7, ASC T1M1, ECMA, Multimedia Communications Forum, CEN/ISSS (Information Society Standardization System) and ICT-SB (Information and Communications Technology Standards Board).

5.4 TINA

5.4.1 Overview

The first TINA Workshop assessed the common need for improving the way services are designed and the common opportunity for tomorrow's services offering, according to increasing customer demands in the telecom community. It was also discovered that similar studies on a software architecture were being conducted in many parts of the world. These lead to the creation of the TINA Consortium in 1993 for cooperatively defining a new software architecture that capitalizes on the latest advances in computer and telecommunication technologies to rationalize the organization of complex software for services and network management.

Towards the end of 1996, the TINA Consortium entered a phase of consolidating its results with the goal of making TINA real in the shortest possible time. Major progress was seen in different areas, e.g., extension of the Service Architecture to new classes of services, consolidation of the Network Resource Architecture and adoption of many features of the DPE architecture by the industry. TINA-C has been interacting with standards bodies and industry consortia, including ATMF, DAVIC, ITU-T, TMF and OMG in order to achieve harmony of mutual specifications and avoid duplication of work [w6].

5.4.2 Description

5.4.2.1 Architecture developed in TINA

TINA Specification covers several architectural frameworks. They are introduced below.

TINA business model

TINA Business model is a high-level enterprise model well applicable to a variety of multimedia info-communication services/businesses. This model provides a set of business roles and a set of reference points where Object interfaces between the interacting roles will be given.

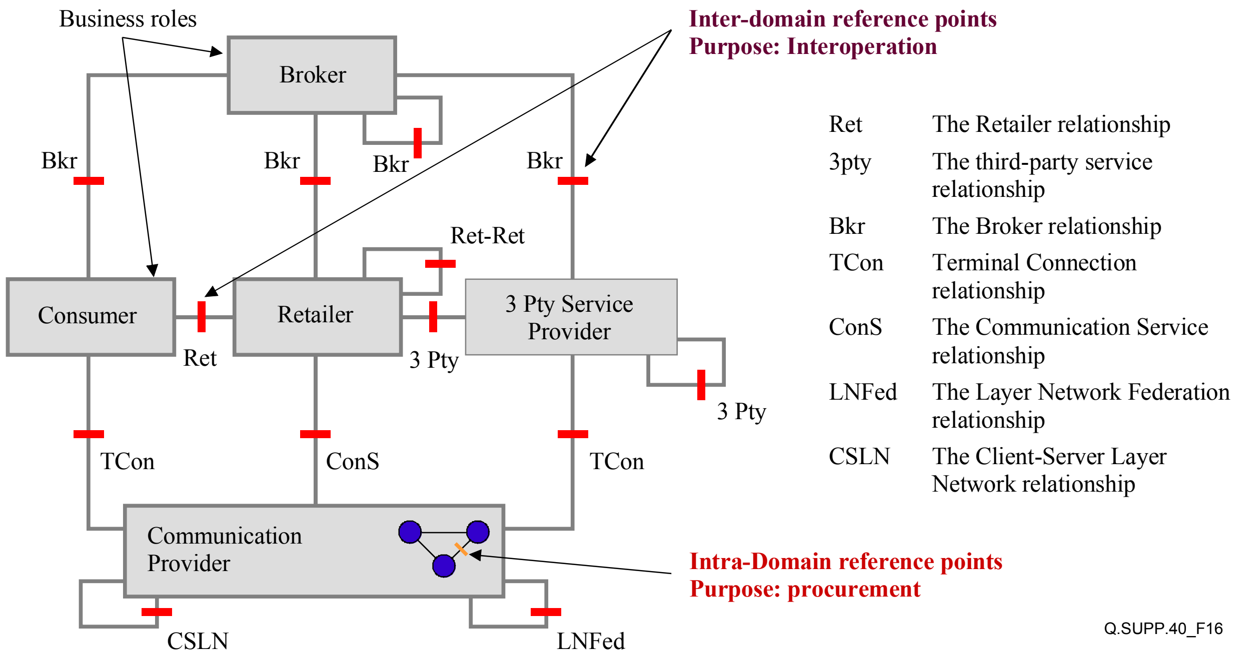

Figure 16 – TINA business model and Reference Points (RPs)

TINA service architecture

TINA Service Architecture provides information and computational models for those generalized objects to be deployed in realizing Access Session and Service Session for multimedia info-communication services/ businesses on top of Communication Session.

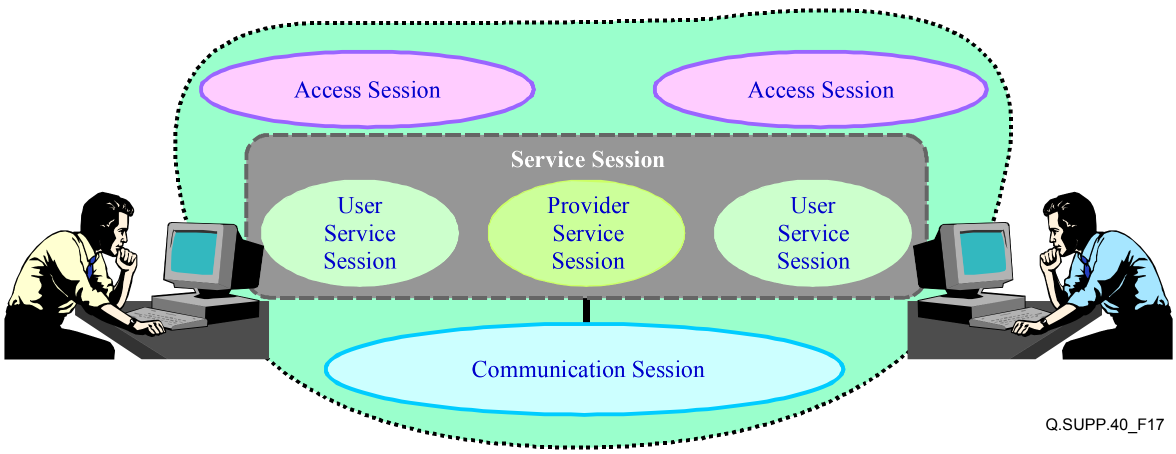

Figure 17 – TINA service architecture

TINA Service Architecture provides a basis for developing Service Architecture Components which may be used in many variation of multimedia services allowed by the business relationship according to TINA Business model. Typical Service Architecture Components are shown below. Here, Object interactions are shown by solid lines between components, and Object interfaces are shown by boxes at the edge of Components in association with related Object interactions.

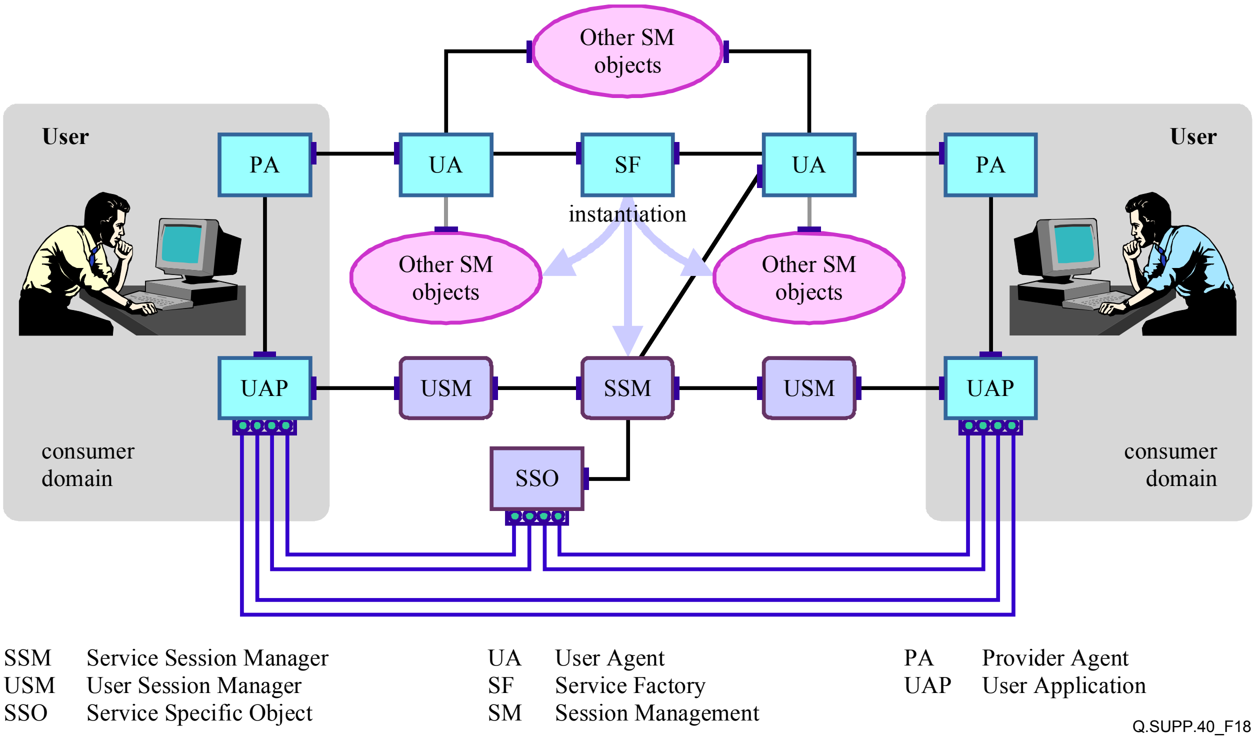

Figure 18 – TINA service architecture components

5.4.2.2 Retailer reference point

Among the TINA specifications related Service Architecture, "Retailer Reference Point" has received highest interest and been reflected in many other standards developed outside of TINA. It provides a specification of the service-independent interface between a Consumer role and a Retailer role as given in the TINA Business Model. Especially, its specification relating to TINA Access Session provides a comprehensive description of APIs on service access and subscription management, and it has been reflected in the TSAS specification standardized by OMG.

5.4.2.3 IN TINA adaptation unit

For the smooth migration from IN to TINA, TINA-C specified "IN access to TINA services and Connection Management (IN-TINA Adaptation Unit)". Through this specification, communication between a legacy IN element and a TINA system is supported. To be more specific, the Adaptation Unit based on this specification is supposed to cope with INAP protocol conversion and adaptation of IN call model in order to ensure communication with a TINA system based on a DPE. In this specification, it will be assumed that the DPE will support operations defined in IDL. An overview of the Adaptation Unit is shown in Figure 19.

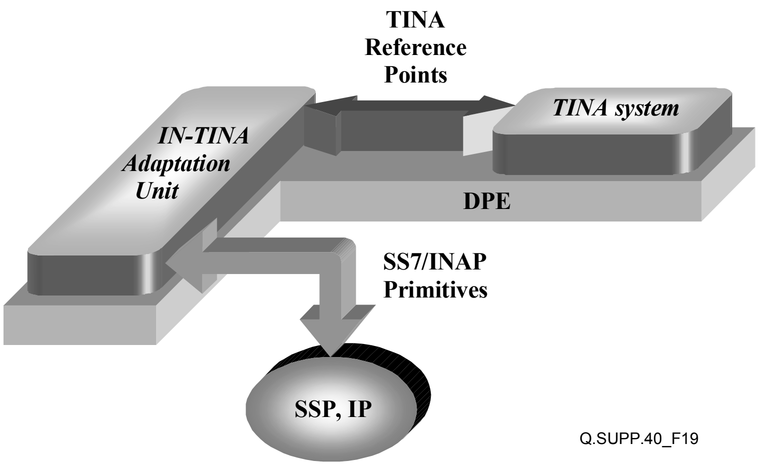

Figure 19 – IN-TINA adaptation unit

5.4.3 Released specification

Modelling and Service-oriented Specifications given by TINA are listed below.

1) Service Architecture;
2) Retailer Reference Point;
3) Service Component Specifications, Computational Model and Dynamics;
4) Business Model and Reference Points;
5) IN access to TINA services and Connection Management (IN-TINA Adaptation Unit);
6) TINA Conformance and Testing Framework.

Note that Network Resource-oriented Specifications established by TINA also exist, including Network Resource Architecture, Network Resource Information Model, Connectivity Service Reference Point, Terminal Connection Reference Point, Network Component Specification and IP Control and Management Specifications, although they are outside the scope of this Supplement.

5.4.4 Schedule

TINA-C completed its intended objectives and terminated its activity at the end of the year 2000. All of its outputs are available in its website [d8].

5.4.5 Relationship with other bodies

TINA-C had been collaborated with related standards bodies including ITU-T, OMG and 3GPP. Several standards reflecting TINA-C specifications have been approved by these bodies, including ITU-T Recs Z.130 and Z.600, OMG's Telecom Service Access and Subscription facility (TSAS) and 3GPP Open Service Architecture (OSA).

6 Applicability

6.1 Objective

It is important to clarify the applicability of each existing API specification, which has already been released or is being studied by standardization organization/industrial forums outside of ITU-T for the following reasons:

– To avoid duplication of API specification work on the same area/target; and
– To provide information to assist the industry in applying API technologies.

Therefore, the objective of this clause is to provide the applicability of each API specification so that an overview of API-related work is given.

6.2 Classification of API

To clarify the applicability of many existing APIs specified by several standardization organization/industrial forums (SDOs, Forums), it is desirable to establish a simple reference model representing related network and application functions. This Supplement utilizes the reference model described in Scope (clause 1) as a base to discuss the applicability of APIs.

In this reference model, the APIs are classified into four categories; Service Control APIs, Service Management APIs, 3rd Party APIs and Protocol/Resource APIs. Protocol/Resource APIs are not within the scope of this Supplement.

6.3 Applicability

This clause provides the applicability of each API/Object Interface.

6.3.1 Criteria for applicability description

This Supplement clarifies the applicability of each API from the viewpoints below.

1) *Functionality*

   The category to which each API described above belongs will be described. This clarification may also provide the information that which APIs are useful for:

   – 3rd Party or

   – Network operator.

   Whether each API/Object Interface is applicable to both fixed and mobile environment or not may also be described.

2) *Language dependency and development support*

   Language dependency/independency, which is closely related to the efficiency of the development, will be described if relevant information is available. Certified development tools, testing tools or reference programs for each API that supports development will also be described if relevant information is available.

3) *Applicability for distributed environment*

   Applicability of each API to the distributed object technologies (e.g. CORBA, RMI, etc) will be described.

6.3.2 Applicability description

6.3.2.1 Parlay/ETSI/3GPP

1) *Functionality*

   Parlay/OSA Framework APIs provide applications with basic mechanisms (e.g. Authentication) that enable them to make use of the service capabilities in the network. 3rd Party applications can access to network capabilities through them when the applications run on the network. Parlay/OSA Framework APIs can be categorized into 3rd Party APIs.

   Parlay/OSA Service APIs for each Service Capability Features (SCF) enable applications to be run by using network capabilities. They can be categorized into Service Control APIs. Network operators will provide them to application software vendors including 3rd Party vendors.

2) *Language dependency and development support*

   OSA API's are technology independent and the Unified Modelling Language (UML) is used as the means to specify class and state transition diagrams.

The Object Management Group's (OMG) Interface Definition Language (IDL) is used as a means to programmatically define the interfaces. IDL files are either generated manually from class diagrams or by using a UML tool.

3) *Applicability for distributed environment*

OSA API specification includes IDL description so that CORBA, RMI etc can be applied for distributed environment.

6.3.2.2 JAIN

6.3.2.2.1 Common technologies

JAIN Common APIs provide some common basic interfaces and classes such as data types and exception definition.

6.3.2.2.2 Specific technologies

1) *Functionality*

JCAT and JCC APIs are call control APIs so they can be categorized into Service Control APIs.

JSLEE APIs provide execution environment and management functions such as deployment. They can be categorized into Service Management APIs.

JSCE APIs provide the environment to create portable telecommunication services on JAIN Service Logic Execution Environment or some other SLEEs. They can be categorized into Service Management APIs.

SPA APIs are the Java technology based implementation of Parlay APIs, so SPA APIs from Parlay Framework APIs can be categorized into 3rd Party APIs. SPA APIs from Parlay Service APIs can be categorized into Service Control APIs.

2) *Language dependency and development support*

JAIN APIs are based on Java technology. In the process of producing JAIN API, Reference Implementation (RI) and Test Compatibility Kit (TCK) are developed for each API and there is certification process for JAIN compliant software.

3) *Applicability for distributed environment*

RMI that is the distributed object technology for java is applicable for JAIN APIs, though JAIN APIs themselves do not take distributed environment into account.

6.3.2.3 OMG

6.3.2.3.1 Common technologies

Unified Modelling Language (UML) provides flexible means to be used in system analysis and design, and has been applied in the basic design of APIs covered in this Supplement. Model Driven Architecture (MDA), based on the UML and other standards for analysis and design, addresses the complete life cycle of designing, deploying, integrating, and managing applications as well as data using open standards, and MDA-based standards enable organizations to integrate any existing systems with those being, and those to be, built in the future. IDL provides means to specify interfaces of objects, and it has been applied to a number of APIs covered in this Supplement. OMG CORBA, along with Inter-ORB communication (GIOP/IIOP) and CORBA Component Model, provides specification of flexible platform for deploying distributed objects.

6.3.2.3.2 API-specific technology

1) *Functionality*

"TSAS" provides authenticated access to capabilities provided by network providers

Core segment defines the interfaces used in the initial phase between different domains including authentication. These interfaces can be categorized into 3rd Party APIs.

Service access segments define the interfaces for inter-domain access. These APIs can be categorized into Service Management APIs.

Subscription segments define interfaces for subscriber-related information and service contract-related. These APIs can be categorized into Service Management APIs.

"CORBA/TC Interworking and SCCP-Inter ORB Protocol" provides interworking between CORBA-based Applications with legacy IN infrastructure, as well as transport of GIOP messages between CORBA-based IN Applications over SS7 Networks by using connectionless SCCP.

2) *Language dependency and development support*

"TSAS" is specified in OMG IDL.

"CORBA/TC Interworking and SCCP-Inter ORB Protocol" provides mapping between IDL and TC/ROS messages as well as mapping between GIOP messages and connectionless SCCP.

3) *Applicability for distributed environment*

"TSAS" is specified based on CORBA so that it can be applied for distributed environment.

"CORBA/TC Interworking and SCCP-Inter ORB Protocol" provides interworking between CORBA-based applications operating in distributed environment.

6.3.2.4 TINA

6.3.2.4.1 Common technologies

TINA Business Model provides a flexible means to model a variety of multimedia services/systems and has served as a basis for defining some APIs covered by this Supplement. TINA Service Architecture provides service-independent decomposition of multimedia service functionality in terms of distributed objects based on the TINA Business Model. TINA Object Description Language (ODL) provides means to specify interfaces of objects, and it has been applied to TINA-based APIs.

6.3.2.4.2 API-specific technology

1) *Functionality*

"Retailer Reference Point" specification provides service management APIs.

"IN TINA Adaptation Unit" specifies conversion between INAP protocol interface and TINA object interface.

2) *Language dependency and development support*

"Retailer Reference Point" is specified by using TINA ODL.

"IN TINA Adaptation Unit" is based on TINA ODL and INAP protocol interface.

3) *Applicability for distributed environment*

"Retailer Reference Point" is specified for distributed environment, named as Distributed Processing Environment (DPE) in TINA. OMG CORBA is used as a typical platform realizing DPE.

6.3.3 Classification of individual APIs/object interfaces

Table 2 – Classification of individual APIs/object interfaces

Standards bodies	Service control APIs	Service management APIs	3rd Party APIs
Parlay/ETSI/3GPP	Service APIs		Framework APIs
JAIN	JCC, JCAT, SPA Service APIs	JSLEE, JSCE	SPA Framework APIs
OMG		TSAS	
TINA	Service Session	Access Session	

Appendix I

This appendix provides the information about released specifications and a schedule of each standards body available at the time of publication of this Supplement. For the latest information, please refer to the appropriate website.

I.1 Parlay/ETSI/3GPP

I.1.1 Released specifications

The API specification developed jointly by the Parlay group, ETSI and 3GPP can be summarized as follows:

1) Overview;
2) Common Data Definition;
3) Framework;
4) Call Control SCF;
5) User Interaction SCF;
6) Mobility SCF;
7) Terminal Capabilities SCF;
8) Data Session Control;
9) Generic Messaging SCF [1];
10) Connectivity Manager SCF [1];
11) Account Management SCF;
12) Charging SCF;
13) Policy Management SCF;
14) Presence and Availability Management (PAM).

The API specifications above are released as Parlay 3.0-3.2 (the Parlay group), ES 201 915 (ETSI) and 3G TS 29.198 (3GPP).

I.1.2 Schedule

The latest information of the API-related activities in Parlay ETSI and 3GPP can be accessed by the following URL.

[1] Not included in the API specifications released by 3GPP

Parlay: http://www.parlay.org/specs/index.asp

ETSI: http://portal.etsi.org/Portal_Common/lite/home.asp

3GPP: http://www.3gpp.org/TB/cn/cn5/specs.htm

I.2 JAIN

I.2.1 Released specifications

Final Release

- JAIN Call Control (JCC) 1.1 : July 2002;
- JAIN user Location and Status 1.0: February 2002.

Proposed Final Draft

- JAIN Service Provider API (SPA) Trust & Security Management and Service Discovery 1.0: August 2001
- JAIN Service Logic Execution Environment (JSLEE) 1.0 : August 2002.

I.2.2 Schedule

The latest information of the API-related activities in JAIN can be accessed by the following URL.

http://java.sun.com/products/jain/api_specs.html

I.3 OMG

I.3.1 Released specifications

1) Specifications on Common Technologies released include:
   - CORBA/IIOP [CORBA3.0];
   - Minimum CORBA [v1.0];
   - Real-Time CORBA [v1.1];
   - Fault-Tolerant CORBA in CORBA/IIOP 3.0;
   - Interface Description Language (IDL) [ISO/IEC 14750:1999] and its Language Mappings;
   - CORBA Component Model (CCM) [v3.0];
   - Object Management Architecture (OMA);
   - Model Driven Architecture (MDA);
   - Unified Modelling Language (UML) [UML 1.4];
   - Meta Object Facilities (MOF)[MOF 1.3];
   - XML Metadata Interchange (XMI) [XMI 1.1];
   - Common Warehouse Metamodel (CWM) [CWM 1.0].
2) Released Application-wise Specifications include:
   - Telecom Service Access and Subscription facility;
   - CORBA/INTC Interworking and SCCP-Inter ORB Protocol.

I.3.2 Schedule

The latest information of the API related activities in OMG can be accessed by the following URL:

http://www.omg.org/news/meetings/tc/Tech_Adoption/index.htm

I.4 TINA

I.4.1 Released specifications

Modelling and Service-oriented Specifications given by TINA are listed below.

1) Service Architecture;
2) Retailer Reference Point;
3) Service Component Specifications, Computational Model and Dynamics;
4) Business Model and Reference Points;
5) IN access to TINA services and Connection Management (IN-TINA Adaptation Unit);
6) TINA Conformance and Testing Framework.

Note that Network Resource-oriented Specifications established by TINA also exist, including Network Resource Architecture, Network Resource Information Model, Connectivity Service Reference Point, Terminal Connection Reference Point, Network Component Specification and IP Control and Management Specifications, although they are outside the scope of this Supplement.

I.4.2 Schedule

TINA-C completed its intended objectives and terminated its activity at the end of the year 2000. All of its outputs are available on its website [w6].

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	**Switching and signalling**
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Geneva, 2003