

International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.2234

(09/2008)

SERIES Y: GLOBAL INFORMATION
INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS
AND NEXT-GENERATION NETWORKS

Next Generation Networks – Service aspects: Service
capabilities and service architecture

Open service environment capabilities for NGN

Recommendation ITU-T Y.2234



ITU-T Y-SERIES RECOMMENDATIONS
**GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-
GENERATION NETWORKS**

GLOBAL INFORMATION INFRASTRUCTURE

General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899

INTERNET PROTOCOL ASPECTS

General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899

NEXT GENERATION NETWORKS

Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Numbering, naming and addressing	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899

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

Recommendation ITU-T Y.2234

Open service environment capabilities for NGN

Summary

Recommendation ITU-T Y.2234 describes open service environment (OSE) capabilities for NGN. OSE capabilities are built on NGN release 1 capabilities with the aim to enable enhanced, flexible service creation and provisioning. This framework requires use of standard interfaces to ensure NGN OSE-based service reusability, portability across networks, as well as accessibility by NGN application providers and/or developers. OSE capabilities enable applications development using NGN OSE-based services for exposing the NGN capabilities.

This Recommendation provides service requirements, functional requirements and architecture of open service environment capabilities.

Source

Recommendation ITU-T Y.2234 was approved on 12 September 2008 by ITU-T Study Group 13 (2005-2008) under Recommendation ITU-T A.8 procedure.

Keywords

Next generation networks (NGN), NGN OSE-based service, open service environment, open service environment capabilities, OSE, service-oriented architecture.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). 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 Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation 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 Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2009

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

	Page
1 Scope	1
2 References.....	1
3 Definitions	1
3.1 Terms defined elsewhere	1
3.2 Terms defined in this Recommendation.....	2
4 Abbreviations and acronyms	2
5 Conventions.....	4
6 Service requirements for open service environment capabilities	4
7 Functional requirements for open service environment capabilities	5
7.1 Service coordination functional requirements.....	5
7.2 Service discovery functional requirements	6
7.3 Service registration functional requirements.....	6
7.4 Service management functional requirements.....	6
7.5 Service composition functional requirements	7
7.6 Service development support functional requirements	7
7.7 Interworking with service creation environment functional requirements.....	7
7.8 Policy enforcement functional requirements.....	8
8 Service architecture for open service environment capabilities	8
8.1 OSE positioning within the NGN reference architecture	8
8.2 Service architecture	10
9 Security considerations.....	12
Appendix I – Use cases for NGN open service environment	13
I.1 Introduction	13
I.2 Service registration and service discovery	13
I.3 BookItWell	14
I.4 IPTV new customer application	16
Appendix II – Relationship of SOA concepts with open service environment capabilities	20
Appendix III – Relationship of standards in other SDOs with open service environment capabilities	21
III.1 Open Service Access (OSA)/Parlay	21
III.2 Open Mobile Alliance (OMA)	21
III.3 Organization for the Advancement of Structured Information Standards (OASIS).....	21
III.4 World Wide Web Consortium (W3C).....	22
III.5 Object Management Group (OMG)	22
III.6 TeleManagement Forum (TMF).....	23
III.7 Information sources for OSE capabilities.....	24
Bibliography.....	28

Introduction

The ability to implement new functionalities in current networks may be limited or impossible due to the capabilities of the installed equipment. Software provisioning to implement new functionalities is essentially restricted to equipment vendors, since the existing or current application programming interfaces (APIs) are typically proprietary (i.e., not open) (Recommendation ITU-T Y.2201). NGN enables new capabilities and supports a wide range of emerging services, including services with advanced and complex functionalities. In response to a drive from application providers and/or developers to develop new applications and capabilities accessible via standard interfaces, there is an increasing need for NGN providers to cooperate in the development of standard application network interfaces (ANIs). Furthermore, software reusability and portability is encouraged to facilitate cost-effective development.

The support of an open service environment within NGN aims to provide efficient and flexible capabilities based on the use of standard interfaces to NGN applications, thereby enabling applications to take full advantage of the NGN capabilities.

Some general benefits of an open service environment are as follows (Recommendation ITU-T Y.2201).

- Applications and capabilities can be easily developed by NGN providers as well as by application providers.
- Capabilities can be made portable and/or reusable across networks.
- Standard ANIs can accommodate interactions between NGN entities and applications.

Within an open service environment, each capability is required to function either independently or in conjunction with other capabilities for the realization of applications. Each capability performs all corresponding service functions for the requesting entity. Applications may be provisioned over different networks, so the capabilities must be able to function independently from the underlying network technologies (Recommendation ITU-T Y.2201).

Recommendation ITU-T Y.2234

Open service environment capabilities for NGN

1 Scope

The objective of this Recommendation is to define an open service environment (OSE) framework for NGN. This framework is built on NGN release 1 capabilities [ITU-T Y.2201] with the aim to enable enhanced, flexible service creation and provisioning.

The use of standard interfaces will ensure NGN OSE-based service reusability, portability across networks, as well as accessibility by NGN application providers and/or developers.

This Recommendation provides service requirements, functional requirements and architecture of open service environment capabilities.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- [ITU-T M.3060] Recommendation ITU-T M.3060/Y.2401 (2006), *Principles for the Management of Next Generation Networks*.
<<http://web.itu.int/rec/T-REC-M.3060>>
- [ITU-T Y.2012] Recommendation ITU-T Y.2012 (2006), *Functional requirements and architecture of the NGN release 1*.
<<http://web.itu.int/rec/T-REC-Y.2012>>
- [ITU-T Y.2201] Recommendation ITU-T Y.2201 (2007), *NGN release 1 requirements*.
<<http://web.itu.int/rec/T-REC-Y.2201>>
- [ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.
<<http://web.itu.int/rec/T-REC-Y.2701>>

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 application network interface (ANI) [ITU-T Y.2012]: Interface which provides a channel for interactions and exchanges between applications and NGN elements. The ANI offers capabilities and resources needed for the realization of applications.

3.1.2 application programming interface (API) [b-ITU-T I.312]: An API provides a set of interfaces from an application environment to an execution environment. The execution environment provides services to the application environment.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 open service environment capabilities: Capabilities provided by an open service environment to enable enhanced and flexible service creation and provisioning based on the use of standards interfaces.

NOTE – Open service environment capabilities enable services reusability, portability across networks and accessibility by application providers and user applications in NGN.

3.2.2 service chain: A set of coordinated services that occur in a specific sequence.

3.2.3 service composition: Service composition is the capability of creating new services from other existing services.

NOTE – NGN OSE-based services will be used as key building blocks to quickly create a composite service. Service composition can occur in a static or in a dynamic way. While in static composition, composite services are defined in advance, dynamic composition sends the request for service discovery using the service description to find the needed services and composes the services during run time.

3.2.4 service coordination: The ability to manage the relationships and interactions among services to provide a "service chain", as well as among services and applications.

3.2.5 service development support: Service development support provides an environment for service creation, development and maintenance.

3.2.6 service discovery: Service discovery performs a search against all registered services and provides the related service information.

3.2.7 service management: Service management provides the overall service managing functions such as service monitoring, service tracking and unexpected failure control.

3.2.8 service registration: Service registration manages the information about services and maintains the most up-to-date information.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ANI	Application Network Interface
API	Application Programming Interface
APL-GW-FE	Application Gateway Functional Entity
APL-SCM-FE	Application Service Coordination Manager Functional Entity
ASF	Application Support Functions
ASF&SSF	Application Support Functions and Service Support Functions
AS-FE	Application Support Functional Entity
BPDM	Business Process Definition Metamodel
BPEL	Business Process Execution Language
BPMN	Business Process Modelling Notation
BPRI	Business Process Run-time Interface
BSS	Billing Support System
CAMEL	Customized Application for Mobile network Enhanced Logic
CDL	Choreography Description Language

DPE	Device Profile Evolution
EAI	External Authoring Interface
ebXML	electronic business eXtensible markup language
EDOC	Enterprise Distributed Object Computing
EJB	Enterprise JavaBeans
eTOM	enhanced Telecom Operations Map
FE	Functional Entity
GPM	Global Management Program
GPS	Global Positioning System
HDTV	High Definition Television
IMS	Internet protocol Multimedia Subsystem
IN	Intelligent Network
INAP	Intelligent Network Application Protocol
IP	Internet Protocol
IPTV	Internet Protocol Television
MOF	Meta Object Facility
MTNM	Multi-Technology Network Management
MTOSI	Multi-Technology Operations System Interfaces
NGN	Next Generation Network
NGOSS	New Generation Operations Systems and Software
NNI	Network-to-Network Interface
OSA	Open Service Access
OSE	Open Service Environment
OSPE	Open Mobile Alliance Service Provider Environment
OSS	Operations Support System(s)
OSS/J	Operations Support System through Java initiative
OWSER	Open Mobile Alliance Web Service Enabler
P2P	Peer-to-Peer
PEEM	Policy Evaluation, Enforcement and Management
QoS	Quality of Service
RAS	Reusable Asset Specifications
RIM	Registry Information Model
S-CSC-FE	Serving Call Session Control Functional Entity
SCA	Service Component Architecture
SCF	Service Control Functions
SCP	Service Control Point
SEC_CF	Security Common Function

SID	Shared Information/Data
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SS-FE	Service Switching Functional Entity
SSF	Service Support Functions
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
UPMS	Unified Modelling Language Profile and Metamodel for Services
VoD	Video on Demand
WIN	Wireless Intelligent Network
WSDL	Web Services Description Language
WSDM	Web Services Distributed Management
XML	eXtensible Markup Language

5 Conventions

None.

6 Service requirements for open service environment capabilities

This clause describes the service requirements of open service environment capabilities for NGN. The NGN open service environment is required to:

- Provide standard APIs for application providers and/or developers to create and introduce applications quickly and seamlessly.
NOTE – Requirements for the standard APIs specified in this Recommendation are at the control level and they do not address media information such as voice and video.
- Provide service level interoperability among different networks, operating systems and programming languages (e.g., web services are an example of enabling technology for providing service level interoperability).
- Support service independence from NGN providers and manufacturers [ITU-T Y.2201]:
 - Functionalities, operations and management of applications are recommended to be all independent from the underlying NGN providers' technologies and infrastructures.
 - Multi-vendor implementations of NGN open service environment are recommended to be supported.
- Support OSE capabilities based on NGN providers' capabilities. However, OSE capabilities based on application providers' capabilities are not supported in this version of this Recommendation.
- Support location, network and protocol transparency [ITU-T Y.2201]:
 - In a distributed environment, application providers and/or developers are recommended to be able to access NGN OSE-based services from anywhere, regardless of their physical location.
 - NGN open service environment capabilities are recommended to allow services to be technology- and terminal-agnostic.

- Protocol transparency is required to be achieved by providing standard protocol programming interface tools for realizing independent service control processes and shielding complex network technical details to the open service environment [ITU-T Y.2201].
- Provide capabilities for coordinating services among themselves and services with applications.
- Support service discovery capabilities to allow users and their devices to discover the services, applications, and other network information and resources of their interest [ITU-T Y.2201]. In addition, discovery mechanisms for services or components of multiple application providers are recommended to be provided.
- Provide the means to manage the registration of capabilities, services and applications. The technology choice is required to ensure functions for service registration and deregistration, including configuration, activation, publication [ITU-T Y.2201].
- Provide the following service management capabilities:
 - service tracking;
 - update management;
 - auditing;
 - version control;
 - logging, e.g., provide a record of the history of services;
 - access control management;
 - statistical analysis of service registration and utilization.
- Support NGN service reuse by providing service composition capability.
- Support of a service composition language.
- Offer a development support environment which supports construction, trialling, deployment and removal of applications [ITU-T Y.2201].
- Allow interworking with service creation environments and network entities for creation and provisioning of applications and services [ITU-T Y.2201].
- Provide a secure access to the NGN capabilities in alignment with the general NGN security requirements as specified in [ITU-T Y.2201].
- Support policy enforcement capability for resource protection and management, and service personalization.

7 Functional requirements for open service environment capabilities

This clause describes functional requirements specific to the support of NGN open service environment capabilities. OSE capabilities consist of service coordination, service discovery, service registration, service management, service composition, service development support, interworking with service creation environments and policy enforcement.

7.1 Service coordination functional requirements

The NGN service coordination functions are required to:

- Provide coordination of applications and services with capabilities [ITU-T Y.2201].
- Provide the tracking of NGN capabilities or service components from various application providers, and the relationship between these capabilities or service components (adapted from [ITU-T Y.2201]).

- Support the information on state change of capabilities or service components for applications and services [ITU-T Y.2201].

7.2 Service discovery functional requirements

The NGN service discovery functions are required to:

- Provide service discovery for physically distributed NGN services.
- Support a variety of discovering criteria (e.g., specific field-based discovery, classification system-based discovery). An example of discovering criteria is implemented in the universal description, discovery and integration (UDDI) specification of web services framework.
- Use user and device profile information for discovering the correct service.
- Allow users to discover user-interest services, device-interest services and network information.
- Support a variety of scoping criteria (e.g., location and cost) to provide appropriate scaling, with appropriate mechanisms to ensure security and privacy (this allows support of customized discovery for a wide range of scenarios).
- Use a variety of approaches for discovering services such as client-server, P2P, combination of client-server and P2P.
- Support appropriate mechanisms to ensure security and privacy [ITU-T Y.2201].
- Take into account scalability (e.g., broadcast mechanisms are recommended to be avoided) [ITU-T Y.2201].

7.3 Service registration functional requirements

The NGN service registration functions are required to:

- Provide service registration, including configuration, activation, publication and service deregistration.
- Provide a variety of service registration features (e.g., manual, autonomous) for NGN services.
- Support a variety of registration parameters, including mandatory and optional parameters.

The NGN service registration functions may support:

- Registration services in centralized and de-centralized ways.
- Multiple concurrent service registrations.

7.4 Service management functional requirements

The NGN service management functions are required to:

- Provide a monitoring function of registered services for availability and predicted response time. NGN services and user applications might need to use monitoring information for the availability or predicted response time of target services before executing services.
- Provide managing functions of QoS information about registered NGN services such as accessibility, performance, integrity, reliability, etc.
- Provide a version management function to NGN services for interoperability.
- Provide notification service functions for updated services.
- Provide failure detection and recovery functions for unexpected failures.

- Provide service tracking management functions to capture and log all relevant information for each component within a service chain. Service tracking is recommended to allow for an association among the captured data associated with a specific service. Service tracking is required to enable tracking of capabilities or components of multiple third parties, and the relationships between these capabilities or components.
- Provide a service substitution function that considers various kinds of factors to users. It is required to provide mechanisms to capture a set of information including terminal capability, network situation, user preference and substitution policy; and judge whether to substitute the service or not based on the captured information. If there is a need to substitute the service, this function will substitute it.
- Provide service access control functions to control the accessibility of a specific service by applications (the service access control function provides the necessary authentication and authorization actions required to ensure that the application has appropriate access rights for the requested service).
- Provide statistical analysis functions to analyse service registration and utilization information (e.g., number of registered services, utilization frequency of registered services and number of applications using registered services).
- Provide an auditing function to review the overall operations of open service environment capabilities during a specific period required by the auditor.

NOTE – The relationship with requirements in [ITU-T M.3060] needs to be considered in the future.

7.5 Service composition functional requirements

The NGN service composition functions are required to:

- Provide a composition language that describes the interaction among services. Additionally, the composition language is recommended to support expression capabilities for describing the composition logic among services.
- Support the composition of services statically or dynamically (i.e., for the static type, the services are composed during service design; while for the dynamic type, the services are composed during service runtime).

7.6 Service development support functional requirements

The NGN service development support functions are required to:

- Support service re-use and allow for service interchangeability [ITU-T Y.2201].
- Support mixing-and-matching of services by management of interfaces and consistent semantics of shared data/schema across these services [ITU-T Y.2201].
- Support the full life cycle of services, ranging from installation, configuration, administration, publishing, versioning, maintenance and removal [ITU-T Y.2201].
- Support delivery-agnostic application design to allow applications to be implemented without requiring re-design for each subsequent development scenario [ITU-T Y.2201].
- Support tracking of dependencies among services [ITU-T Y.2201].

7.7 Interworking with service creation environment functional requirements

The NGN service creation environment interworking functions are required to:

- Support the following three classes of service creation environments [ITU-T Y.2201]:
 - Open service creation environment (examples of this class of environment using ANI include OSA/Parlay, Parlay X, OMA).
 - IP multimedia subsystem (IMS)-based service creation environment.

- Intelligent network (IN)-based service creation environment (examples of relevant interface protocols for this class of environment include IN application protocol (INAP), customized application for mobile network enhanced logic (CAMEL) and wireless intelligent network (WIN)).

7.8 Policy enforcement functional requirements

The NGN policy enforcement functions are required to:

- Provide a description language to express various types of policy rules such as those related to authorization, charging, service level agreement and logging. This language is recommended to support policy re-use.
- Provide a policy execution framework to interpret and execute the policies.
- Protect services from unauthorized users' requests and manage requests based on the policy rules.
- Support the selection of appropriate services for service composition to respond to the needs and preferences of a user or a group of users.

8 Service architecture for open service environment capabilities

This clause describes service architecture for the support of open service environment capabilities in NGN. This architecture is based on the NGN release 1 functional architecture [ITU-T Y.2012].

8.1 OSE positioning within the NGN reference architecture

Figure 1 is an extension of Figure 1 in [ITU-T Y.2012]. It shows an extended NGN architecture enabling support of open service environment capabilities.

The functional group identified as "OSE" in Figure 1 provides an open service environment for NGN and enables NGN applications to implement enhanced services that make use of NGN capabilities. The OSE functional group is positioned inside the application support functions and service support functions of the NGN service stratum.

NOTE – Applications in Figure 1 are provided by various application providers and/or developers and are not intended to be limited to a single application provider or developer.

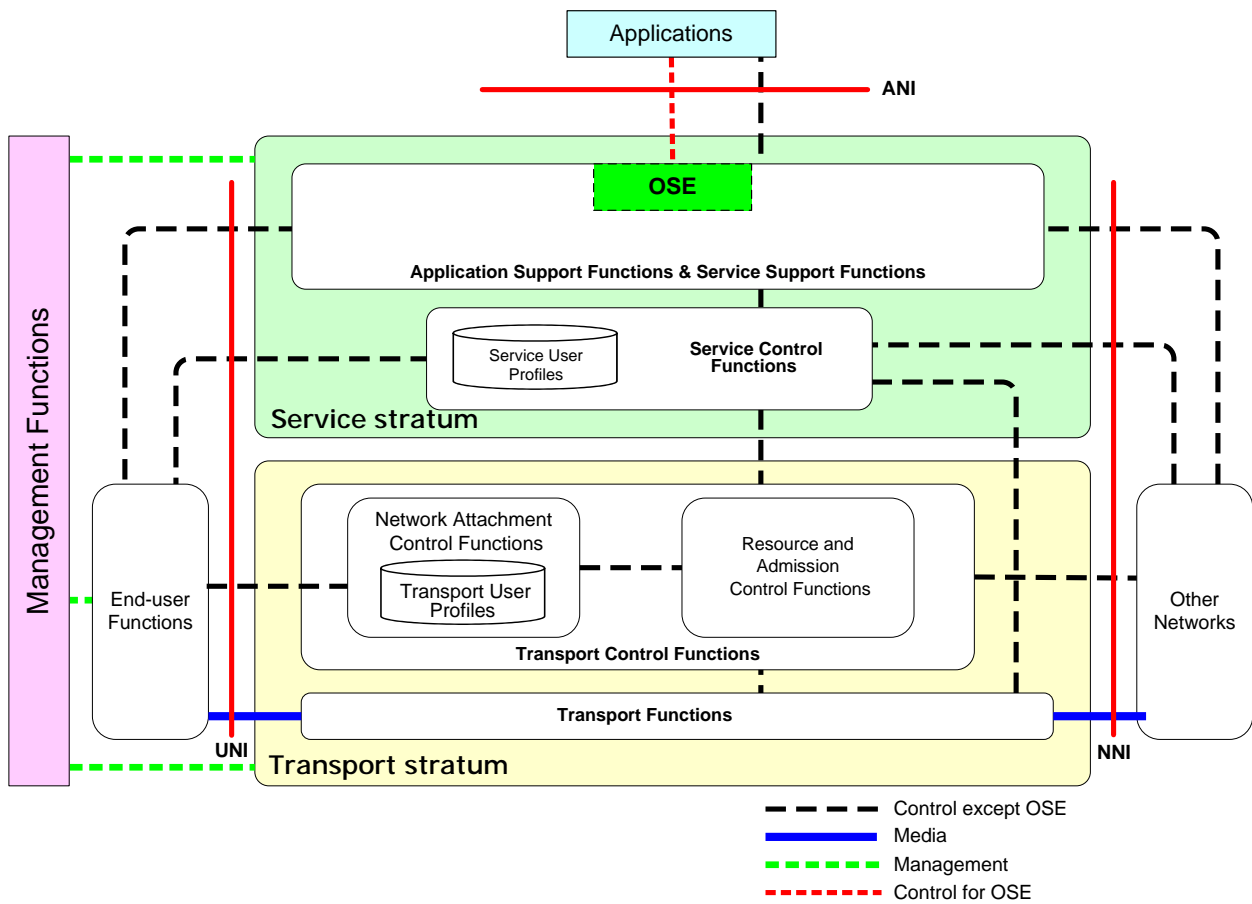


Figure 1 – Extended NGN architecture positioning the OSE

"Control for OSE" in Figure 1 represents the control level interfaces between applications and OSE. These interfaces do not support exchange of media information such as voice and video.

"Control except OSE" in Figure 1 represents the interactions and exchanges between applications and NGN services which are external to the interactions and exchanges between applications and OSE.

Figure 2 shows the functional components constituting the OSE functional group: service coordination, service discovery, service registration, service management, service composition, service development support, interworking with service creation environments, and policy enforcement.

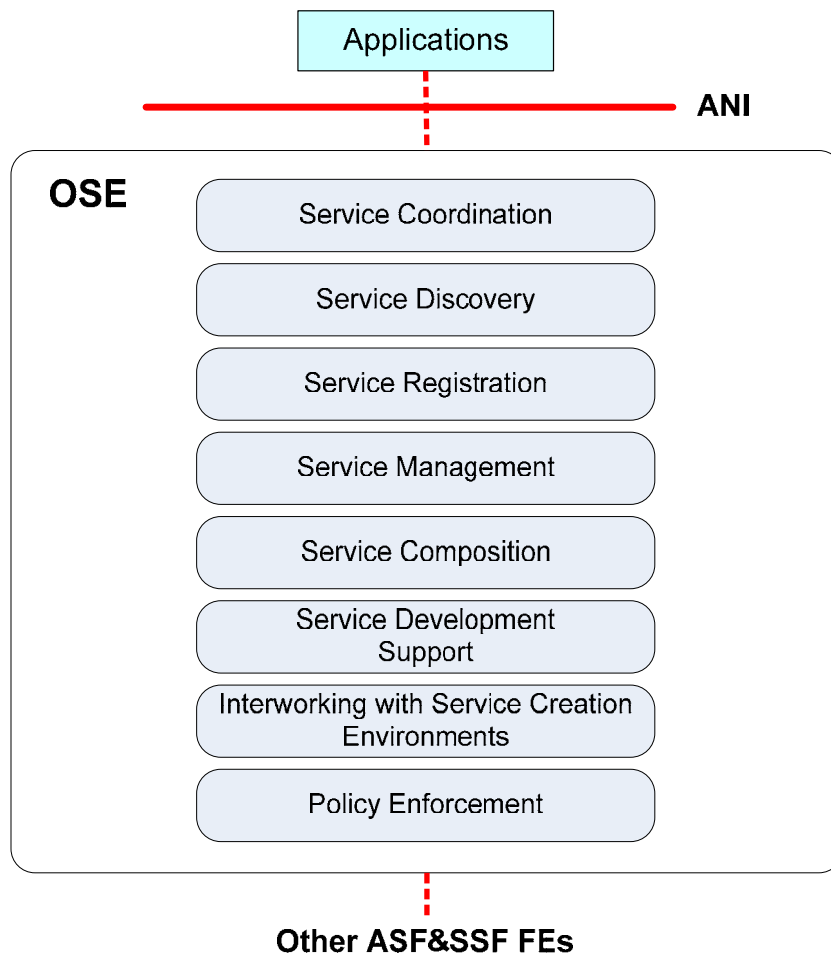


Figure 2 – Functional components of the OSE functional group

8.2 Service architecture

This clause provides details of the service architecture related to the OSE functional group.

Figure 3 is an extended NGN architecture positioning the OSE with an expanded view of the OSE functional group.

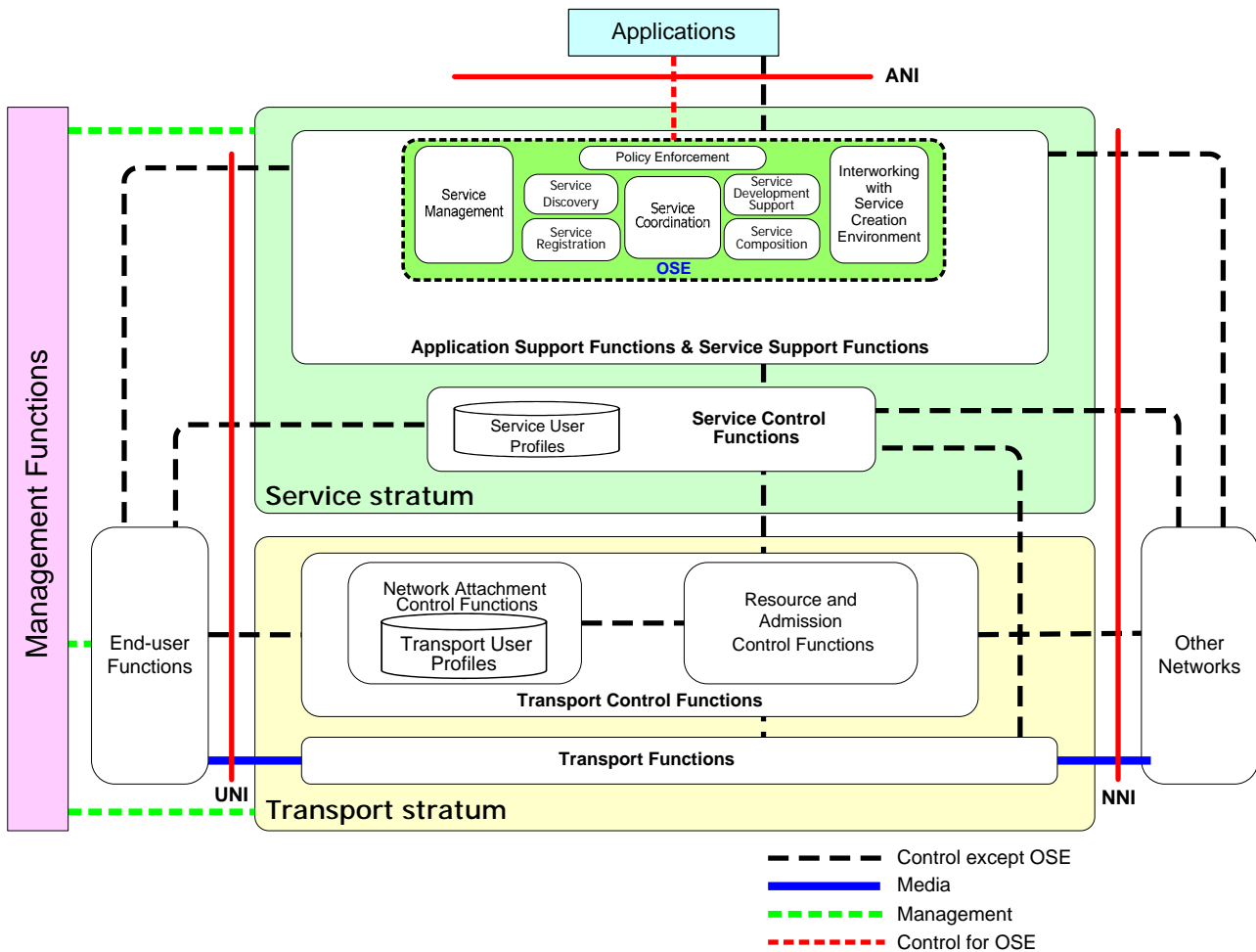


Figure 3 – Extended NGN architecture positioning the OSE with expanded view of the OSE functional group

8.2.1 Relationship between OSE functional components and functional entities of the NGN release 1 functional architecture

OSE functional components use standard interfaces to NGN applications and user services enabling applications to be implemented taking full advantage of NGN capabilities.

According to [ITU-T Y.2012], the application gateway functional entity (APL-GW-FE) in ASF&SSF (application support functions and service support functions) is the interworking entity between various functions of NGN and all external application servers and service enablers. The application service coordination manager functional entity (APL-SCM-FE) manages interactions between multiple application services (or servers). And the functional entities of ASF&SSF might interwork with each other via APL-SCM-FE to provide services to the end users.

NOTE – The relationship of OSE service management functions and management functions in Figure 3 needs to be defined and described in the future. In so doing, the principles for the management of NGN in [ITU-T M.3060] should be applied.

Table 1 provides a set of candidate options for the mapping of OSE functional components into [ITU-T Y.2012] ASF&SSF functional entities.

**Table 1 – Mapping of OSE functional components into [ITU-T Y.2012]
ASF&SSF functional entities**

	APL-GW-FE	APL-SCM-FE	AS-FE	SS-FE	New FE (currently not identified)
Mapping of OSE functional components into ASF&SSF FEs	Serves as an interworking entity between the applications, and services and capabilities of the NGN (adapted from [ITU-T Y.2012])	Manages interactions between multiple application services (or servers) [ITU-T Y.2012]	Supports generic application server functions including hosting and executing services [ITU-T Y.2012]	Provides access and interworking to a legacy IN SCP [ITU-T Y.2012]	
Service coordination	Not applicable	Optional	Not applicable	Not applicable	Optional
Service discovery	Optional	Not applicable	Not applicable	Not applicable	Optional
Service registration	Optional	Not applicable	Not applicable	Not applicable	Optional
Service management	Optional	Not applicable	Not applicable	Not applicable	Optional
Service composition	Not applicable	Optional	Not applicable	Not applicable	Optional
Service development support	Optional	Not applicable	Not applicable	Not applicable	Optional
Interworking with service creation environments	Optional	Not applicable	Optional	Optional	Optional
Policy enforcement	Optional	Optional	Not applicable	Not applicable	Optional

9 Security considerations

In the interaction of the open service environment capabilities with NGN applications, it is recommended that the security requirements of [ITU-T Y.2201] and [ITU-T Y.2701] be taken into consideration, including the following dimensions of NGN security: access control, authentication, non-repudiation, data confidentiality, communication security, data integrity, availability and privacy.

Appendix I

Use cases for NGN open service environment

(This appendix does not form an integral part of this Recommendation)

I.1 Introduction

As next generation networks (NGNs) evolve and are deployed, they are expected to enable an open service environment that will foster service creation. This appendix identifies a number of service or service feature use cases whose implementation and deployment would benefit from support of an NGN open service environment.

I.2 Service registration and service discovery

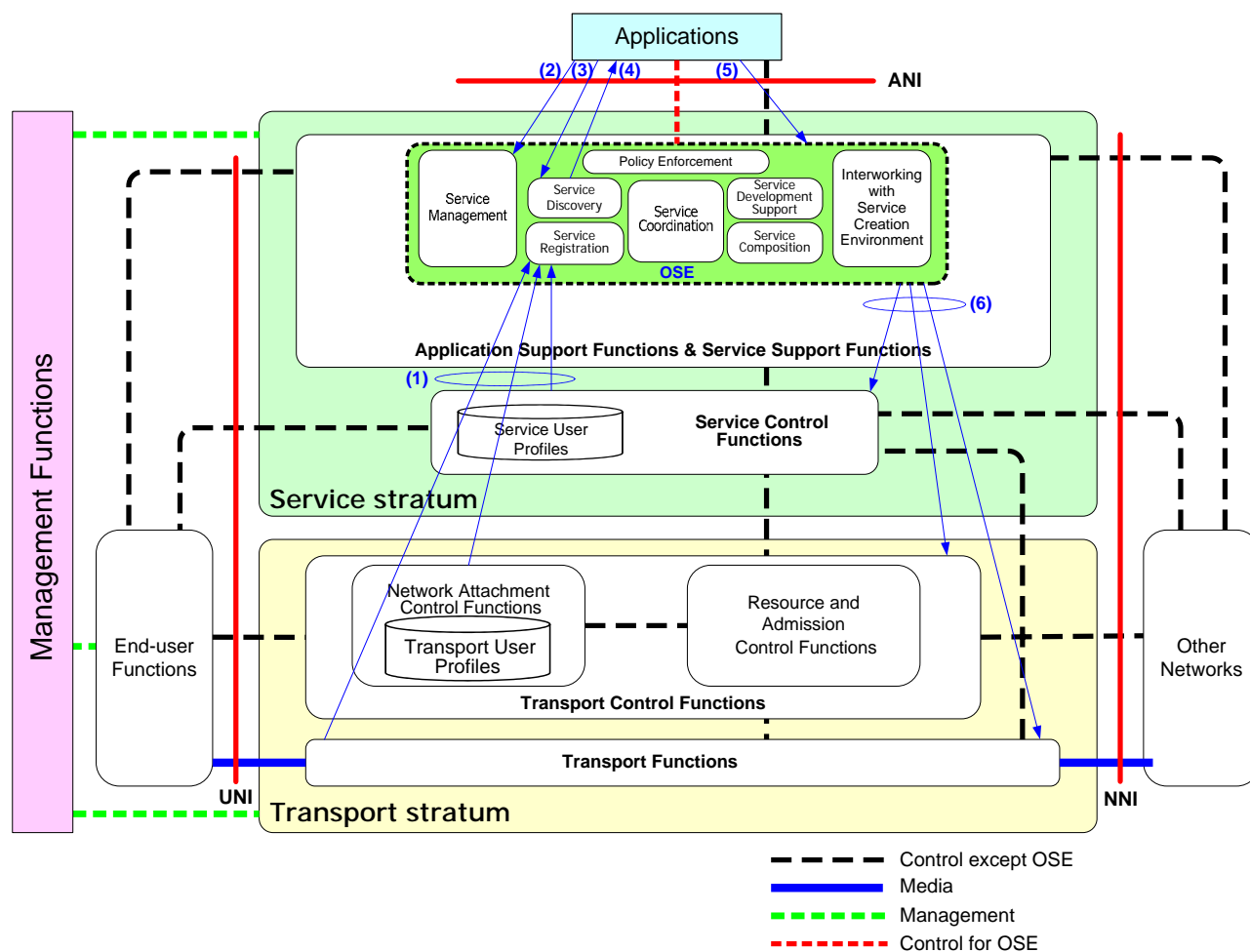


Figure I.1 – Interactions between applications, NGN services and OSE capabilities

This clause describes a use case of how applications can use NGN services with open service environment capabilities. The scenario is outlined in Figure I.1 and it is assumed that the authentication function of the NGN service stratum is provided to the application through the OSE.

The process associated with this use case is the following:

At first, NGN services in both service and transport strata and new services created via service development support are registered (step 1). Next, service registration processes this information and generates information for service discovery and management.

When the application contacts the open service environment capabilities, the application is at first authenticated and authorized from service management to usage of NGN services (step 2). After the application invokes service discovery, it obtains a list of the available services and selects one among them. Then it requests the associated interface information from service discovery (step 3). Service discovery returns the interface (step 4). From this moment onwards, the application is able to use the service (steps 5 and 6).

I.3 BookItWell

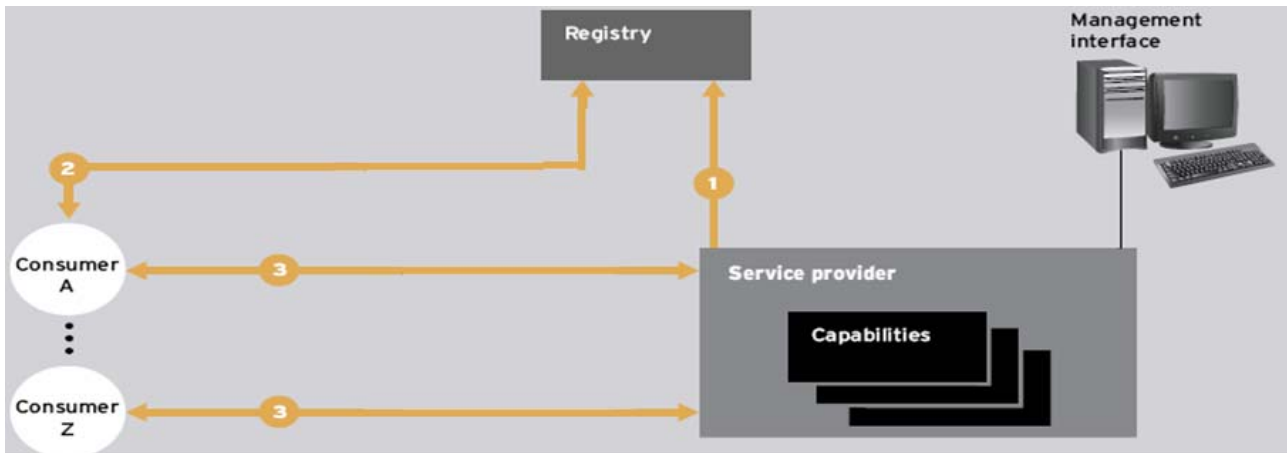


Figure I.2 – Interactions between service provider and service consumers

NGN open service environment comprises several major capabilities that allow potential service consumers to locate and invoke services (i.e., based on OSE service registration and service discovery capabilities). Service providers make their services visible by publishing service descriptions in a registry (1). Typically, the universal description, discovery and integration (UDDI) protocol is used between a service provider and the registry. A potential consumer of the service also uses the UDDI protocol to ask the registry about available services and receives the appropriate service descriptions from the registry (2). These descriptions contain sufficient information to allow the consumer to interact with the service provider to cause the desired real-world effect (3). The syntax of this interface is normally specified using the web services description language (WSDL). A management interface enables the service owner to configure, monitor and take service capabilities through their lifecycles from introduction through depreciation to removal in an orderly manner. This interface may use such protocols as web services distributed management (WSDM) and WS management.

People driving long distances in their car understand the annoyance associated with trying to find a suitable hotel at the end of a long day of travel if they were not able to arrange in advance a place for the evening. Such a search would involve obtaining local maps and driving off the main road to look for hotels with vacancies. Add some small children in the back seat asking "are we there yet?" and the imaginatively named "BookItWell" service provided by "DriveALot Inc." could be the answer (although the service is fictitious, all of the components exist and several related services are already available from various service providers).

As illustrated in Figure I.2, the BookItWell service can be built by combining a number of services provided independently by different companies, including those for hotel booking, weather forecasting, location, instant messaging, route calculation and billing.

Each evening, after taking into account the consumer location, the weather forecast and pre-loaded hotel preferences, the service locates and books a suitable hotel room for the consumer and sends a message, giving directions to the hotel, to his/her cellular telephone, or programs the consumer car's global positioning system (GPS) with the appropriate information.

BookItWell simply buys access to the underlying services. Whether the location service is based on cellular telephone triangulation or GPS is irrelevant to BookItWell as long as the contracted service quality (up-time, accuracy, precision, etc.) is maintained. To allow BookItWell to combine the underlying services easily, it is essential that each service provides details of its interface in a standard format.

BookItWell makes use of two weather forecasting services and chooses the one that is cheaper for a particular request. In this case, the underlying services not only make their interfaces available, but also provide details of their costs and coverage.

With the exception of the weather forecasting service, all the components (i.e., underlying services) of the BookItWell service are composed based on static service composition. Static service composition invokes the service components directly based on composite service logic.

The invocation of a given weather forecasting service is done in a dynamic way. Service composition makes use of service discovery to identify the preferred service component, based on the service request, and then invokes the selected one. In the case of the BookItWell service, the criterion of selection is the cost of the invocation in a specific context. In general, the service request for dynamic service composition includes the description of the required service, and the information can be mapped to more than one exact service component. When a consumer invokes one of these capabilities through an interaction with the provider (link #3 in Figure I.2), some real-world effect occurs. In this use case, when the BookItWell consumer interacts with the hotel booking service, a room is booked – a real-world effect. Note that BookItWell is both a provider (to DriveALot's subscribers) and a consumer (of another provider's route calculation service). A capability together with its specification, contract and real-world effect is defined as a service in this context.

In this use case, a company has developed a route calculation service. This service has a contract, interface specification and sufficient security to ensure that it is used only by authorized consumers.

Providers have to make their services visible to allow potential consumers to discover them (using OSE service registration capability). They do this by publishing (exposing) a service description containing information about three aspects of the service: its behaviour, its interface, and its policies and contracts. In this use case, the two potential suppliers of weather forecasting services advertise the service description of their services, allowing BookItWell to discover them and interact with them.

The interface description includes the specific protocols, commands and information exchange by which actions are initiated that result in real-world effects, as specified through the service functionality portion of the service description. Service providers and service consumers are together known as service participants.

Once a service has been made visible, it can be combined with other visible services to create a higher-level or orchestrated service that typically implements a particular business process.

In this use case, DriveALot has orchestrated (see Appendix II for definition of orchestration) using the OSE service composition capability to combine underlying services into a business flow known as BookItWell. Each of the underlying services that are used may themselves be composed of lower-level services, but this is of no interest to DriveALot.

In addition to orchestration, new higher-level services can be created from simpler ones through choreography (see Appendix II for definition) using the OSE service composition capability. Although there are several differences between orchestration and choreography, the primary

outward difference lies in where the state of the flow is held: in the case of orchestration, it is held in a central controller; in the case of choreography, it is held in each invoked service.

In this example orchestrated service, when the weather-forecasting system is invoked, it may take several seconds to respond. During this time, the state of the composite service is stored centrally so that the response from the weather forecast service is expected when it arrives and the next step in the process can be activated. In a choreographed service, on the other hand, this state information would be carried through each of the various service providers.

The interactions of the type shown in Figure I.3 (invoke service X, and if that fails or if it is, for example, a Tuesday, then invoke service Y, then invoke service Z, and so on) are typically drawn using a graphical editor and then converted into a formal language. In the case of orchestration, they could be converted into business process execution language (BPEL); and in the case of choreography, they could be converted into choreography description language (CDL).

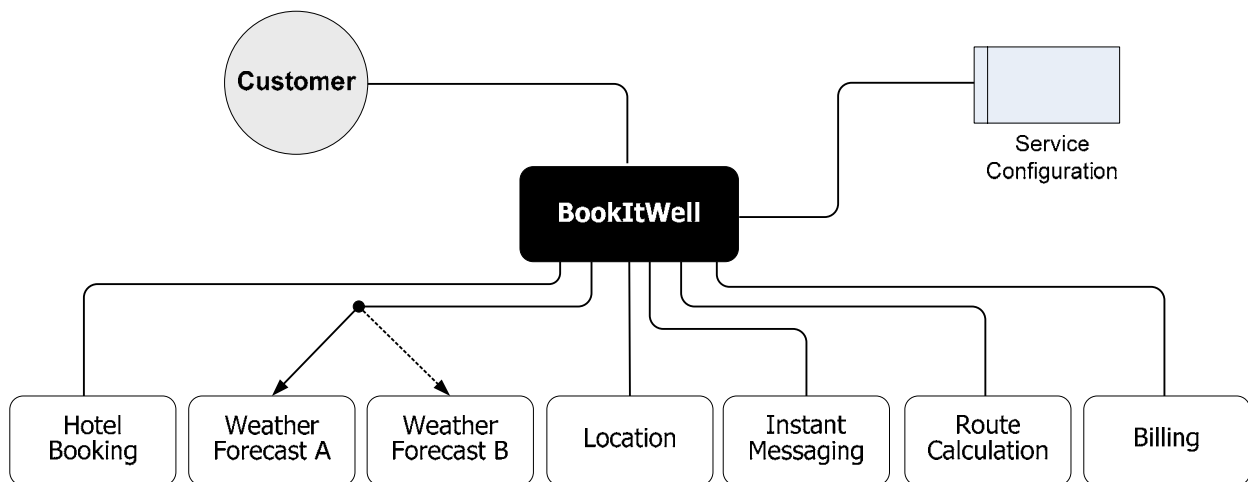


Figure I.3 – BookItWell service example

I.4 IPTV new customer application

This clause provides an IPTV new customer application use case. This use case utilizes a particular IPTV customer service as an example to show how a service could be created in an open environment. This example is telco-centric and shows how complex and sophisticated services (created based on the OSE service development support capability) can be delivered to end users using OSE service composition capability provided by various vendors and components via the OSE service registration capability.

The service can be plotted using a layered architecture. A simplified version of this architecture is shown in Figure I.4 and consists of a component/vendor layer and an OSE service composition capability layer. This layered architecture can be expanded by composing additional OSE service composition layers on top of existing service layers. The services in the higher layer (built using the OSE service composition capability) may be created using a subset and/or abstraction of services from the lower layer(s) (using the OSE service registration).

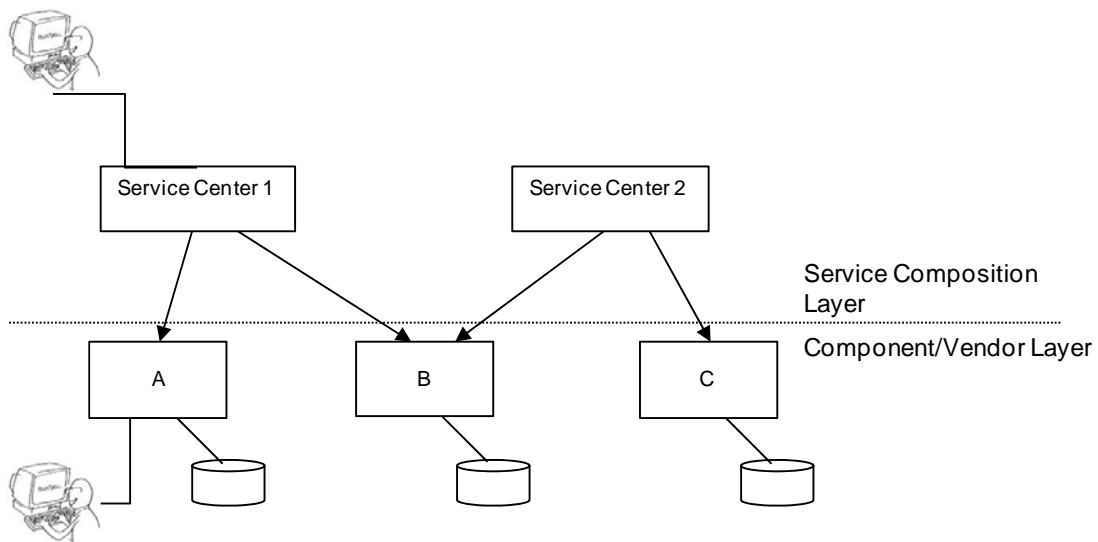
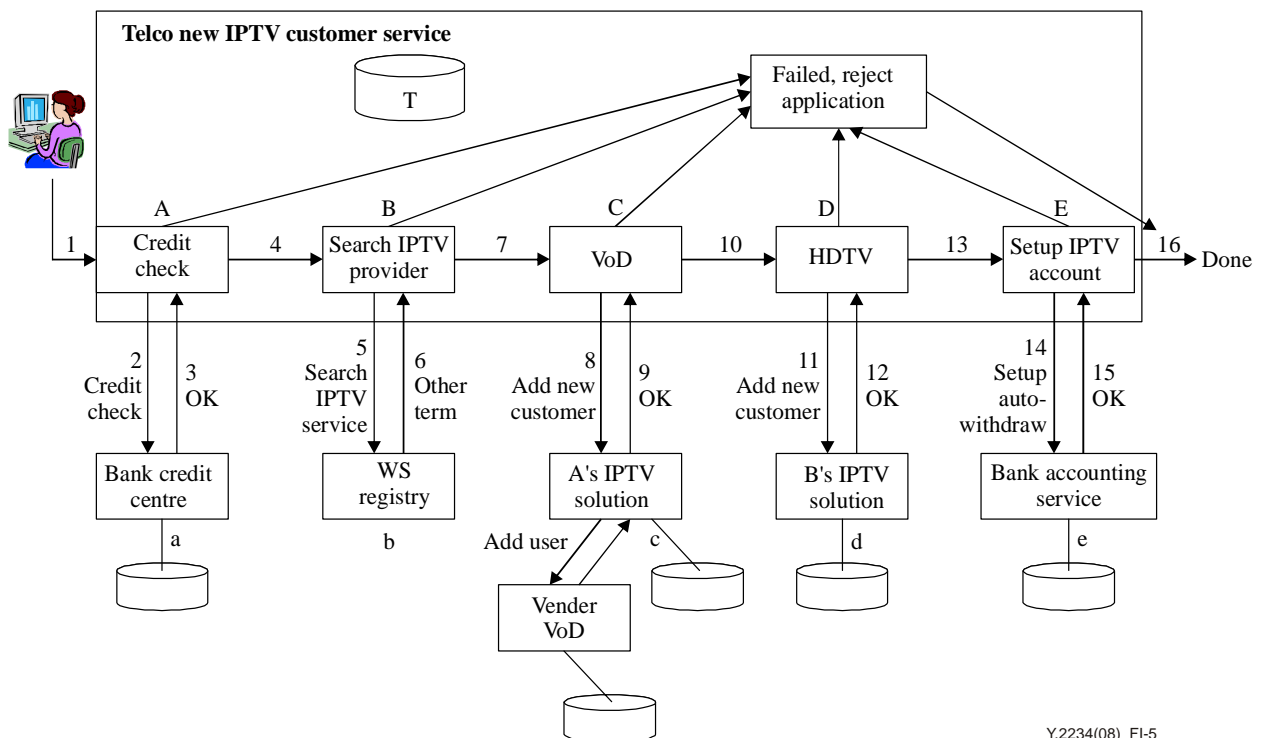


Figure I.4 – Telco service layered architecture

In the architecture shown in Figure I.4, we assume that A, B and C can exchange workflow information, thereby linking the higher-level service with the corresponding device/component models. Using this layered concept as the approach to service development, an IPTV use case is developed.

The use case presented is an IPTV new customer processing service. In this use case, the telco composite service layer creates an IPTV new customer processing service from the underlying services and management functions provided by various IPTV services and financial services. This new customer service is essentially an ordered invocation of a business process which involves several steps: credit check, search for IPTV service providers, set-up vendor-level IPTV services and set-up financial information services.



Y.2234(08)_FI-5

Figure I.5 – IPTV use case

A process used to invoke this service could be as follows:

- 1) A new customer contacts the telco IPTV customer service department to apply for a new IPTV service which includes VoD and HDTV.
- 2) The credit check process contacts the bank credit centre to see if this customer has a good credit standing.
- 3) The credit centre checks its own management database to search for the customer credit data and returns the result of the customer credit check.
- 4) The telco service continues to the next step in the business process when the return status is OK; otherwise, it goes to "Failed, reject application".
- 5) The telco service searches for available VoD and HDTV services.
- 6) The service registry returns a couple of entries – Vendor VoD and company B's HDTV solutions.
- 7) The telco service continues to the next step in the business process when the return status is OK; otherwise, it goes to "Failed, reject application".
- 8) The telco service contacts company A's IPTV service provider to add the new customer and to enable the new service for this customer.
- 9) Company A's IPTV service creates new instances of IPTV customer and services, stores the information in its local management database and returns OK to the telco service.
- 10) The telco service continues to the next step in the business process when the return status is OK; otherwise, it goes to "Failed, reject application".
- 11) The telco service contacts company B, who provides IPTV services, to add a new customer and to enable the new service for this customer.
- 12) Company B's IPTV service creates a new instance of IPTV customer and services, stores the information in its local management database and returns OK to the telco service.
- 13) The telco service continues to the next step in the business process when the return status is OK; otherwise, it goes to "Failed, reject application".
- 14) The telco service contacts the bank accounting service to set-up an auto-withdraw/auto-deposit entry.
- 15) The bank providing the accounting service creates a new entry for this customer and returns OK to the telco service.
- 16) The telco service creates a new entry with a unique customer ID and charging information, and stores necessary vendor-specific customer information in its own local database, including company A's/company B's customer ID and the bank accounting service customer ID/account/transaction requisition number. The new customer set-up for this service is now complete.

The underlying services and the selection of the underlying services are likely to change from time to time. It is imperative that the customer does not become aware of these underlying changes and, when services are properly implemented, these changes are recommended to be transparent from a customer's perspective.

Below are two examples:

- 1) A new service provider publishes the availability of a less expensive or a better service using the OSE service registration capability. The telco service is aware of such a new service deployment via a registry search based on the use of the OSE service discovery capability to obtain the new service.
- 2) A customer would like to choose a different package (i.e., network VoD) via the same telco service provider. The telco service will need to search a fitting service provider and set-up the new service and, in the meantime, will substitute the existing services associated with the underlying services based on the OSE service composition capability.

Appendix II

Relationship of SOA concepts with open service environment capabilities

(This appendix does not form an integral part of this Recommendation)

Service orientated architecture (SOA) is a software architecture concept that uses loosely coupled software services to support the requirements of business processes and software users. Resources on a network in an SOA environment are made available as independent services that can be accessed without the knowledge of their underlying platform implementation.

The definition of SOA, as in [b-OASIS SOA-RM], reads as follow: "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations".

Another definition [b-Newcommer] of SOA is a style of software design that guides all aspects of creating and using business services. SOA is also a way to define and provision an IT infrastructure to allow different applications to exchange data and participate in business processes, regardless of the operating systems or programming language underlying those applications.

In line with the SOA paradigm, OSE requires the use of standardized interfaces to make NGN capabilities accessible and easy to be used by NGN application providers and developers.

In this Recommendation, some terminologies are used in the SOA paradigm, including the following two terms related to service composition:

(web) service orchestration: Service orchestration defines the sequence and conditions in which one (web) service invokes other (web) services in order to realize some useful function, i.e., an orchestration is the pattern of interactions that a (web) service agent must follow in order to achieve its goal [b-W3C Glossary].

service choreography: Service choreography defines the sequence and conditions under which multiple cooperating independent agents exchange messages in order to perform a task to achieve a goal state [b-W3C Glossary].

Appendix III

Relationship of standards in other SDOs with open service environment capabilities

(This appendix does not form an integral part of this Recommendation)

This appendix is intended to identify standards in other SDOs that are related to open service environment capabilities.

III.1 Open Service Access (OSA)/Parlay

OSA/Parlay standards include the basic framework and standard APIs for network capabilities [b-ETSI ES 203 915-3].

Parlay X web service APIs provide a more granular specification than Parlay/OSA APIs. They represent an abstraction and simplification of the Parlay/OSA APIs. Therefore a Parlay X application can be written in any language as long as it can make the proper web service invocations. There are several development tools, for different programming languages, to create, deploy and interact with web services. Parlay X web service APIs may be provided by a Parlay web services gateway which is an intermediary function between the Parlay application server and the Parlay/OSA gateway. However, Parlay X specifications do not mandate a particular implementation approach or architecture and may be used to provide access to service enablers and components through simple abstracted web service APIs.

OSA/Parlay [b-ETSI ES 203 915-3] information related to OSE capabilities is in clause III.7.

III.2 Open Mobile Alliance (OMA)

The OMA is the focal point for the development of mobile service enabler specifications, which support the creation of interoperable end-to-end mobile services. OMA drives service enabler architectures and open enabler interfaces that are independent of the underlying wireless networks and platforms. OMA creates interoperable mobile data service enablers that work across devices, service providers, operators, networks and geographies [b-OMA].

OMA specifies an OMA service environment, which is a flexible and extensible architecture that offers support to a diverse group of application developers and service providers. Enablers of OMA service environments provide standardized components to create an environment in which services may be developed and deployed.

OMA service environment [b-OMA OSE] information related to OSE capabilities is in clause III.7.

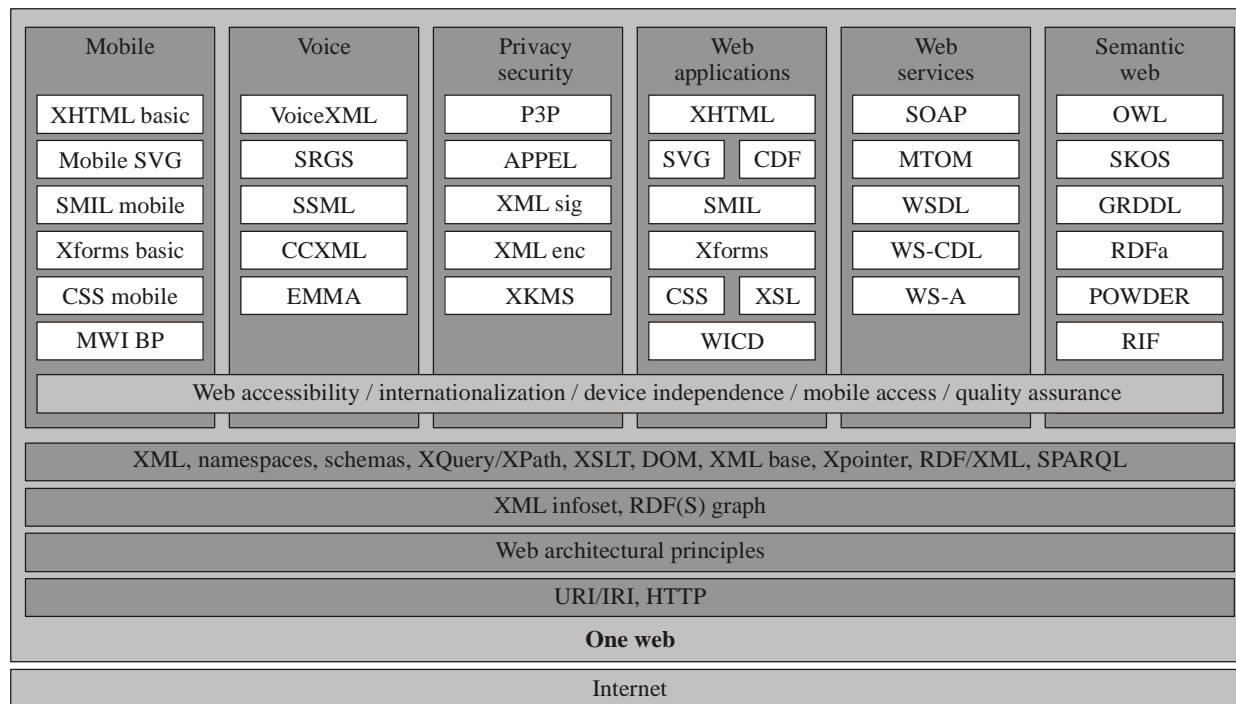
III.3 Organization for the Advancement of Structured Information Standards (OASIS)

OASIS produces web services standards along with standards for security, e-business and standardization efforts in the public sector and for application-specific markets [b-OASIS].

OASIS continues to develop and enhance standards for some of the essential underpinnings of web services, such as managing distributed services, implementing service registry and security, and, at a higher level, for implementing business processes.

OASIS [b-OASIS] information related to OSE capabilities is in clause III.7.

III.4 World Wide Web Consortium (W3C)



Y.2234(08)_FIII-1

Figure III.1 – The W3C technology stack

The World Wide Web Consortium [b-W3C] is the main international standards organization for the worldwide web (abbreviated WWW or W3). Figure III.1 illustrates one view of the web infrastructure, the main focus of W3C.

W3C is developing the web services choreography description language (WS-CDL) that is targeted for composing interoperable, peer-to-peer collaborations between any types of participant regardless of the supporting platform or programming model used by the implementation of the hosting environment.

W3C information related to OSE capabilities is in clause III.7.

III.5 Object Management Group (OMG)

OMG SOA-related activities are based on a platform-independent view. Their focus is the extension of the model driven architecture (MDA) approach, including unified modelling language (UML) to ensure complete and consistent service specifications architected to work together to meet business needs.

These standards cover a wide spectrum of usage, from high-level representations of an organization and its objectives through operational SOA architectures mapping to technology-specific specifications and their implementations. This includes business rules, processes, service specifications as well as the management of systems and assets.

The architectural assets developed in the SOA-MDA process contribute to the entire solutions, life-cycle including business analysis, solutions architecture, service interfaces, acquisition, implementation, testing, publishing, discovery and usage of services.

OMG [b-OMG] information related to OSE capabilities is in clause III.7.

III.6 TeleManagement Forum (TMF)

The TeleManagement Forum (TMF) is an international membership organization of communications service providers and suppliers to the communications industry. While operations support systems (OSS) are generally dominated by proprietary and custom technologies, the TMF is focused on standards and frameworks for OSSs.

The newest stage in OSS architecture work has come with the TMF's NGOSS programme establishing a set of principles that OSS integration is recommended to adopt, along with a set of models that provide standardized approaches. The models include an information model (the Shared Information/Data model, or SID), a process model (the enhanced Telecom Operation Map, or eTOM), an application model (the Telecom Applications Map), an architecture (the Technology Neutral Architecture) and a lifecycle model. The TMF describes NGOSS as an architecture that is:

- "loosely coupled";
- distributed;
- component based;

along with application components upon which a communications service provider business can run.

The components interact through a common communications vehicle (using an information exchange infrastructure, e.g., EAI, web services, EJB).

The behaviour can be controlled through the use of process management and/or policy management to orchestrate the functionality provided by the services offered by the components.

The early focus of the TMF's NGOSS work was on building reference models to support a business stakeholder view on process, information and application interaction. Running in parallel were activities that supported an implementation stakeholder view on interface specifications to provide access to OSS capability (primarily MTNM). The MTNM work evolved into a set of web services providing multi-technology operations system interfaces (MTOSI). Most recently, the OSS, through Java initiative (OSS/J), joined the TMF to provide NGOSS-based BSS/OSS APIs.

Another relevant initiative within TMF is the service delivery framework (SDF) programme, aiming to define the framework that supports and integrates all functions required for the lifecycle of a service delivered to a customer, across all stakeholders in a service provider environment.

TMF [b-TMF] information related to OSE capabilities is in clause III.7.

III.7 Information sources for OSE capabilities

NGN capabilities	OSA/Parlay [b-ETSI ES 203 915-3]	OMA OSE [b-OMA OSE]	OASIS [b-OASIS]	W3C [b-W3C]	OMG [b-OMG]	TMF [b-TMF]
Service coordination		PEEM (policy evaluation, enforcement and management) OSPE (OMA service provider environment)	WS-Coordination WS-Business activity WS-Atomic transaction	Web services policy – Framework Web services policy – Attachment Web services policy namespace Web services policy XML schema	Current effort: – UPMS (SOA extension of UML) – BPDM Existing standards: – UML – EDOC: component architecture – Enterprise distributed object computing	TMF053 series: NGOSS technology neutral architecture (TNA) GB921 series: eTOM, business process framework GB922 series: SID, shared information architecture NGOSS Contract metamodel (work in progress)
Service discovery	Discovery of framework and network service capability features	OWSER (UDDI) OMA's DPE OMA's GPM	Universal description, discovery and integration (UDDI) ebXML registry information model (RIM) ebXML registry services and protocols (RS)	Web services description language (WSDL)	Current effort: – UPMS (SOA extension of UML) – BPDM Existing standards: – RAS: Reusable asset specifications – RAS description: Metamodel for describing and managing reusable assets	

NGN capabilities	OSA/Parlay [b-ETSI ES 203 915-3]	OMA OSE [b-OMA OSE]	OASIS [b-OASIS]	W3C [b-W3C]	OMG [b-OMG]	TMF [b-TMF]
Service management	Registering of network service capability features, integrity management	OSPE (OMA service provider environment)	Management using web services (WSDM-MUWS) Management of web services (WSDM-MOWS) WS-Notification WS-Brokered notification	Service modelling language WS-Eventing	BPRI: Business process runtime interface Description: Looking at runtime system, monitoring and measuring it, and evaluating these measurements against expectations RAS: To publish the services	Current effort: Service delivery framework Description: A framework that supports and integrates all functions required for the lifecycle of a service delivered to customers, across all stakeholders in a service provider environment. SDF unifies under a logical view service design, creation/composition, deployment, activation, provisioning, sale and campaign management, execution, operations, charging, billing and revenue management, retirement, monitoring and trouble resolution, etc.

NGN capabilities	OSA/Parlay [b-ETSI ES 203 915-3]	OMA OSE [b-OMA OSE]	OASIS [b-OASIS]	W3C [b-W3C]	OMG [b-OMG]	TMF [b-TMF]
Service composition		PEEM (policy evaluation, enforcement and management)	Business process execution language for web services	Web services choreography description language	Current effort: – UPMS – BPMN – BPDM	
Service development support		XDM, OSPE (OMA service provider environment)		Service modelling language	Current effort: – UPMS – BPMN – BPDM Existing standards: – EDOC	TMF053 series: NGOSS technology neutral architecture (TNA) GB921 series: eTOM, business process framework GB922 series: SID, shared information architecture GB942 contract guidelines and principles NGOSS contract metamodel MTNM/MTOSI, OSS/J (TIP)
Service registration		OSPE (OMA service provider environment)	ebXML registry information model (RIM) ebXML registry services and protocols (RS) Universal description, discovery and integration (UDDI)		Existing Standards: – RAS – MOF	

NGN capabilities	OSA/Parlay [b-ETSI ES 203 915-3]	OMA OSE [b-OMA OSE]	OASIS [b-OASIS]	W3C [b-W3C]	OMG [b-OMG]	TMF [b-TMF]
Interworking with service creation environments						
Policy enforcement	Policy management SCF	PEEM (policy evaluation, enforcement and management)	Service component architecture (SCA) policy framework Privacy policy profile of XACML	Web services policy – Framework Web services policy – Attachment Web services policy namespace Web services policy XML Schema Web services policy – Primer Web services policy – Guidelines for policy assertion authors		SID policy framework
Security	Authentication, authorization	SEC_CF (security common function)	WS-Security WS-Security: SOAP message security WS-Security: Username token profile WS-Security: SAML token profile WS-Security: X.509 certificate token profile WS-Federation			

Bibliography

- [b-ITU-T I.312] Recommendation ITU-T I.312 (1992), *Principles of intelligent network architecture*.
<<http://web.itu.int/rec/T-REC-I.312>>
- [b-ETSI ES 203 915-3] ETSI ES 203 915-3 (2005), *Open Service Access (OSA); Application Programming Interface (API); Part 3: Framework (Parlay 5)*.
<<http://pda.etsi.org/pda/queryform.asp>>
- [b-Newcommer] Newcommer, E. and Lomow, G. (2004), *Understanding SOA with Web Services*, Addison-Wesley Professional.
- [b-OASIS] OASIS. <<http://www.oasis-open.org/home/index.php>>
- [b-OASIS SOA-RM] OASIS SOA-RM (2006), *Reference Model for Service Oriented Architecture 1.0*.
<<http://www.oasis-open.org/specs/index.php#soa-rmv1.0>>
- [b-OMA] Open Mobile Alliance. <<http://www.openmobilealliance.org>>
- [b-OMA OSE] Open Mobile Alliance OSE 1.0.3 (2006), *OMA Service Environment*.
<http://www.openmobilealliance.org/technical/release_program/ose_v1_0.aspx>
- [b-OMG] The Object Management Group. <<http://www.omg.org>>
- [b-TMF] TeleManagement Forum. <<http://www.tmforum.org>>
- [b-W3C] World Wide Web Consortium. <<http://www.w3.org>>
- [b-W3C Glossary] World Wide Web Consortium Working Group Note (2004), *Web Services Glossary*.
<<http://www.w3.org/TR/ws-gloss>>

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
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	Telecommunication management, including TMN and network maintenance
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, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems