

ITU-T Recommendation Y.2234 (Y.ngn-openenv)

Open service environment capabilities for NGN applications

Summary

This Recommendation describes open service environment (OSE) capabilities for NGN applications. OSE is built on NGN release 1 capabilities [ITU-T Y.2201] with the aim to enable enhanced, flexible service creation and provisioning. This framework makes use of standard interfaces to ensure NGN OSE based services reusability, portability across networks, as well as accessibility by third parties and NGN applications developers. Even though currently NGN provides some standard interfaces, these standard interfaces may not be sufficient to support all future NGN services. OSE enables 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.

Keywords:

Open service environment, Open service environment capabilities, OSE, NGN OSE based service, Service Oriented Architecture, NGN and Next Generation Networks.

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

Table of Contents

1. Scope.....	3
2. References.....	3
3. Definitions.....	4
3.1. Terms defined elsewhere	4
3.2. Terms defined in this Recommendation	4
4. Abbreviations and acronyms.....	5
5. Conventions	6
6. Service requirements for open service environment capabilities	7
7. Functional requirements for open service environment capabilities	8
7.1. Service coordination functional requirements	8
7.2. Service discovery functional requirements.....	8
7.3. Service registration functional requirements	9
7.4. Service management functional requirements.....	9
7.5. Service composition functional requirements	10
7.6. Service development support functional requirements	10
7.7. Interworking with service creation environments functional requirements	10
7.8. Policy enforcement functional requirements	10
8. Service architecture for open service environment capabilities	11
8.1. OSE positioning within the NGN reference architecture	11
8.2. Service architecture	12
9. Security considerations	15
Appendix I	16
Appendix II.....	22
Appendix III	23
III.1 Open Service Access (OSA)/Parlay	23
III.2 Open Mobile Alliance (OMA).....	23
III.3 Organization for the Advancement of Structured Information Standards (OASIS)	23
III.4 World Wide Web Consortium (W3C)	23
III.5 Object Management Group (OMG).....	24
III.6 TMF: The TeleManagement Forum (TMF)	24
III.7 Information sources for OSE capabilities	26
Bibliography	31

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) [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 third party application and service providers to develop new applications and capabilities accessible via standard interfaces, there is an increasing need for network and service providers to cooperate in the development of standard application network interfaces (ANI). Furthermore, software reusability and portability is encouraged to facilitate cost effective development.

While there are some standard interfaces for third party application provider and user applications in NGN, these standard interfaces may not be sufficient to support all future NGN services.

To respond to this industry need, NGN should support an open service environment that will provide efficient and flexible capabilities based on the use of standard interfaces to NGN applications and user services thereby enabling applications to take full advantage of the NGN capabilities.

Some general benefits of an open service environment are as follows [ITU-T Y.2201].

- Applications and capabilities can be easily developed by network providers as well as by third parties.
- 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 shall be able 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 [ITU-T Y.2201].

1. Scope

The objective of this Recommendation is to define an open service environment 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.

Making use of standard interfaces, it ensures NGN OSE based services reusability, portability across networks, as well as accessibility by third parties and NGN applications 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 Y.2012] ITU-T Recommendation Y.2012 (2006), *Functional requirements and architecture of the NGN*

[ITU-T Y.2201] ITU-T Recommendation Y.2201 (2006), *NGN release 1 requirements*

[ITU-T Y.2701] ITU-T Recommendation Y.2701 (2006), *Security requirements for NGN release 1*

3. Definitions

3.1. Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1. Application programming interface [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.

3.1.2. Application network interface [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.

3.2. Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1. Open service environment capabilities: Capabilities provided by 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 third parties and user applications in NGN.

3.2.2. Service coordination: The ability to manage the relationships and interactions among services to provide a “service chain” as well as among services and applications.

3.2.3. Service chain: A set of coordinated services that occur in a specific sequence.

3.2.4. 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 composite service. Service composition can occur in a static or in a dynamic way. While in static composition, composite services are known defined in advance, dynamic composition sends the request for service discovery using the service description to find the needed services and composes the services on the fly.

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

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

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
BPMN	Business Process Modeling Notation
BPDM	Business Process Definition Metamodel
BPEL	Business Process Execution Language
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	Electronic Telecom Operations Map
GPM	Global Management Program
GPS	Global Positioning System
HDTV	High Definition Television
IMS	IP Multimedia Subsystem
INAP	IN Application 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
OASIS	Organization for the Advancement of Structured Information Standards

OMA Open Mobile Alliance
OMG Object Management Group
OSA Open Service Access
OSE Open Service Environment
OSPE OMA Service Provider Environment
OSS Operations Support System(s)
OSS/J Operation Support System Through Java Initiative
OWSER OMA 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
SCA Service Component Architecture
S-CSC-FE Serving Call Session Control Functional Entity
SCF Service Control Function
SEC_CF Security Common Function
SID Shared Information Data
SOA Service Oriented Architecture
SOAP Simple Object Access Protocol
SSF Service Support Function
TMF TeleManagement Forum
UDDI Universal Description, Discovery, and Integration
UML Unified Modeling Language
UPMS UML Profile and Metamodel for Services
VOD Video on Demand
W3C World Wide Web Consortium
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 service providers, third party application developers, and potentially end users 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 include media information such as voice and video.
- Provide the 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 network provider and manufacturers [ITU-T Y.2201]:
 - Functionalities, operations and management of third party service provider applications and value added services should be all independent from the underlying network providers' technologies and infrastructures.
 - Multi-vendor implementations of NGN open service environment should be supported.
- Support location, network and protocol transparency [ITU-T Y.2201]:
 - In a distributed environment, third party service providers should be able to access services from anywhere, regardless of their actual physical location.
 - NGN open service environment should 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 process 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 third parties should be provided. However, independent third party components are not supported inside the OSE;
- 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 services reuse by providing service composition capability;
- Support of a service composition language;
- Offer a development support environment which should support construction, trialing, 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 clause 5.13 of [ITU-T Y.2201];
- Support policy enforcement capability for resources protection and management, and service personalization.

7. Functional requirements for open service environment capabilities

This clause describes functional requirements specific to the support of the 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 service providers, and the relationship between these capabilities or service components [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 Discovery, Description and Integration (UDDI) of Web Services framework;
- Use user and device profile information for discovering the proper 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 should 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 function of QoS information about registered NGN services such as accessibility, performance, integrity, reliability, etc;
- Provide version management function to NGN services for interoperability;
- Provide notification service functions for updated services;
- Provide failure detection and recovering 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 should 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 service substitution function that considers various kinds of factors (e.g. terminal capability, network situation, user preference) 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 function to analyze 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.

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 should 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 services re-use and allow for services 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 designs 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 environments 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 [Q.1741.x];
 - Intelligent network (IN)-based service creation environment.
Examples of relevant interface protocols for this class of environment include IN application protocol (INAP), customised 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 kinds of policy rules such as those related to authorization, charging, service level agreement and logging. This language should support policies 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 reference 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, NGN architecture overview. 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 are not intended to be limited to a single application provider.

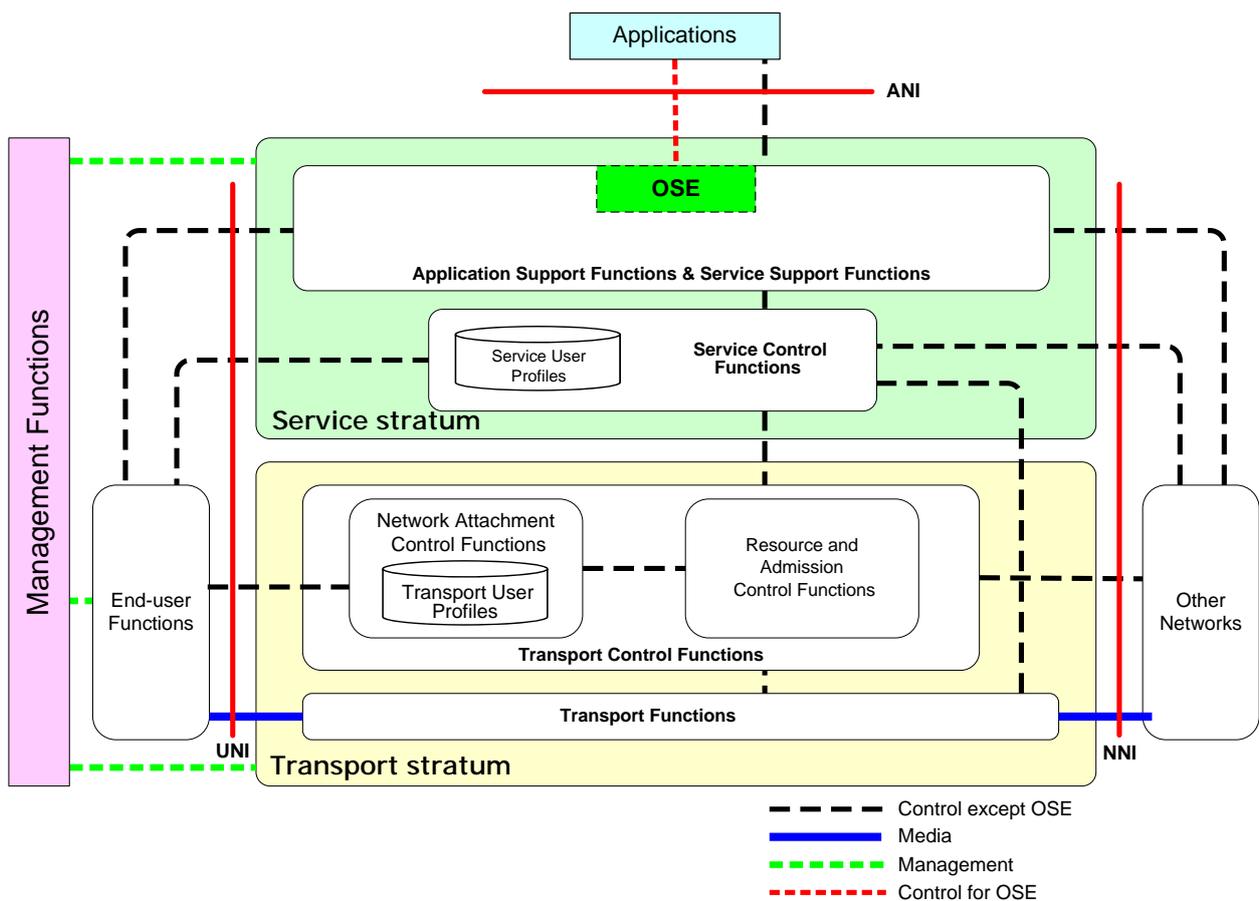


Figure 1 – Extended NGN architecture positioning the OSE

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

“Control for ASF&SSF (except OSE)” 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 development support, service coordination, service discovery, service management, service composition, service registration, interworking with service creation environments and policy enforcement.

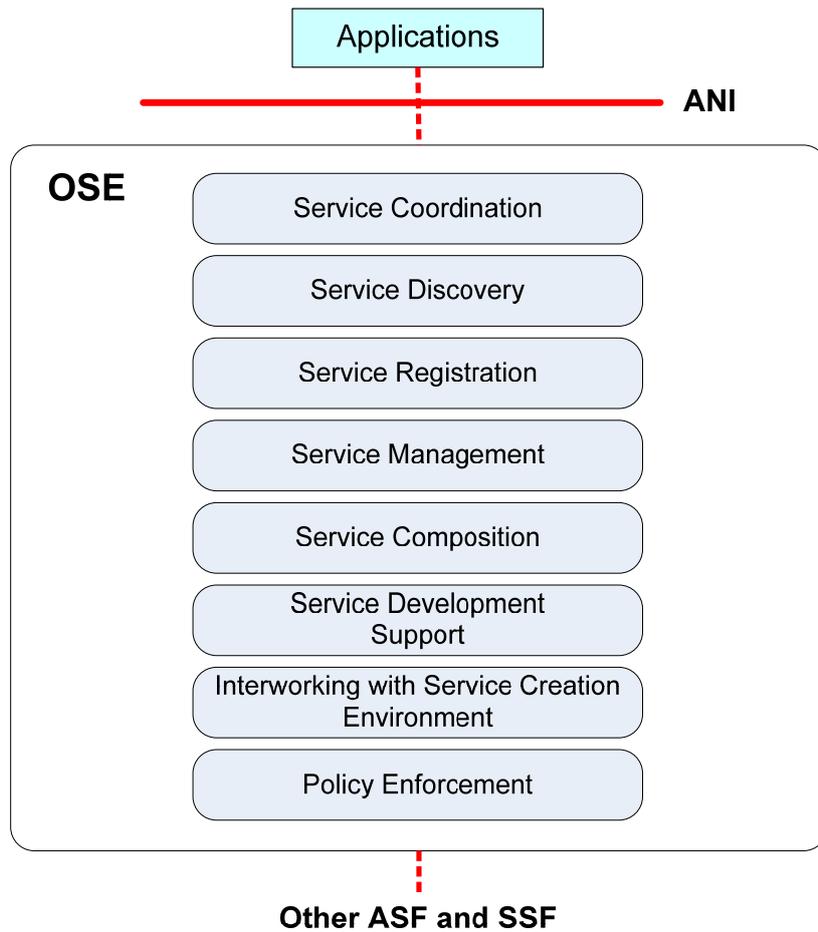


Figure 2 – Functional components of the OSE functional group

8.2. Service architecture

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

Figure 3 is a further extended NGN architecture positioning the OSE.

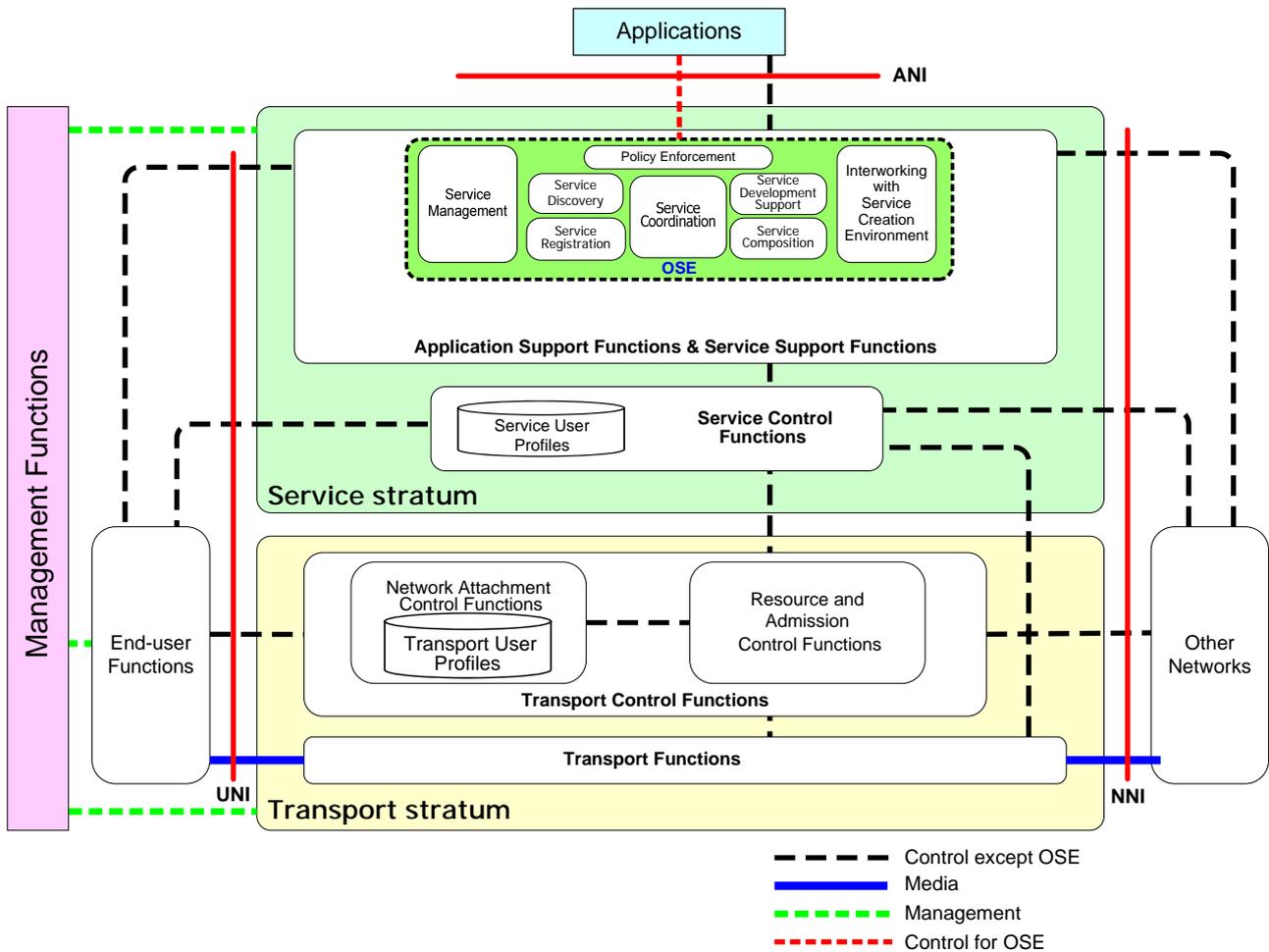


Figure 3 – Further extended NGN architecture positioning the OSE

8.2.1. Relationship between OSE functional components and functional entities of the NGN release 1 reference 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 the ITU-T Recommendation of functional requirements and architecture for the NGN release 1 (Y.2012), 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 convergent services to the end users.

Table 1 provides a mapping between OSE functional components and [ITU-T Y.2012] ASF&SSF functional entities.

Table 1: Mapping between OSE functional components and [ITU-T Y.2012] ASF&SSF functional entities

Mapping between Open Service Environment capabilities and ASF&SSF		Rec.Y.2012 ASF&SSF				
		APL-GW-FE	APL-SCM-FE	AS-FE	SS-FE	New FE
		serves as an interworking entity between the applications and , services and capabilities of the NGN (adapted from Y.2012)	Manages interactions between multiple application services (or servers) [Y.2012]	supports generic application server functions including hosting and executing services [Y.2012]	Provides access and interworking to a legacy IN SCP [Y.2012]	
O S E	Service discovery	optional mapping				optional mapping
	Service management	optional mapping				optional mapping
	Service registration	optional mapping				optional mapping
	Service coordination		optional mapping			optional mapping
	Service composition		optional mapping			optional mapping
	Service development support	optional mapping				optional mapping
	Interworking with service creation environments	optional mapping				optional mapping
	Policy Enforcement	optional mapping	optional mapping			optional mapping

9. Security considerations

In the interaction of the open service environment capabilities with NGN applications, it is recommended to consider security requirements of Y.2201 [ITU-T Y.2201] and Y.2701 [ITU-T Y.2701] 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

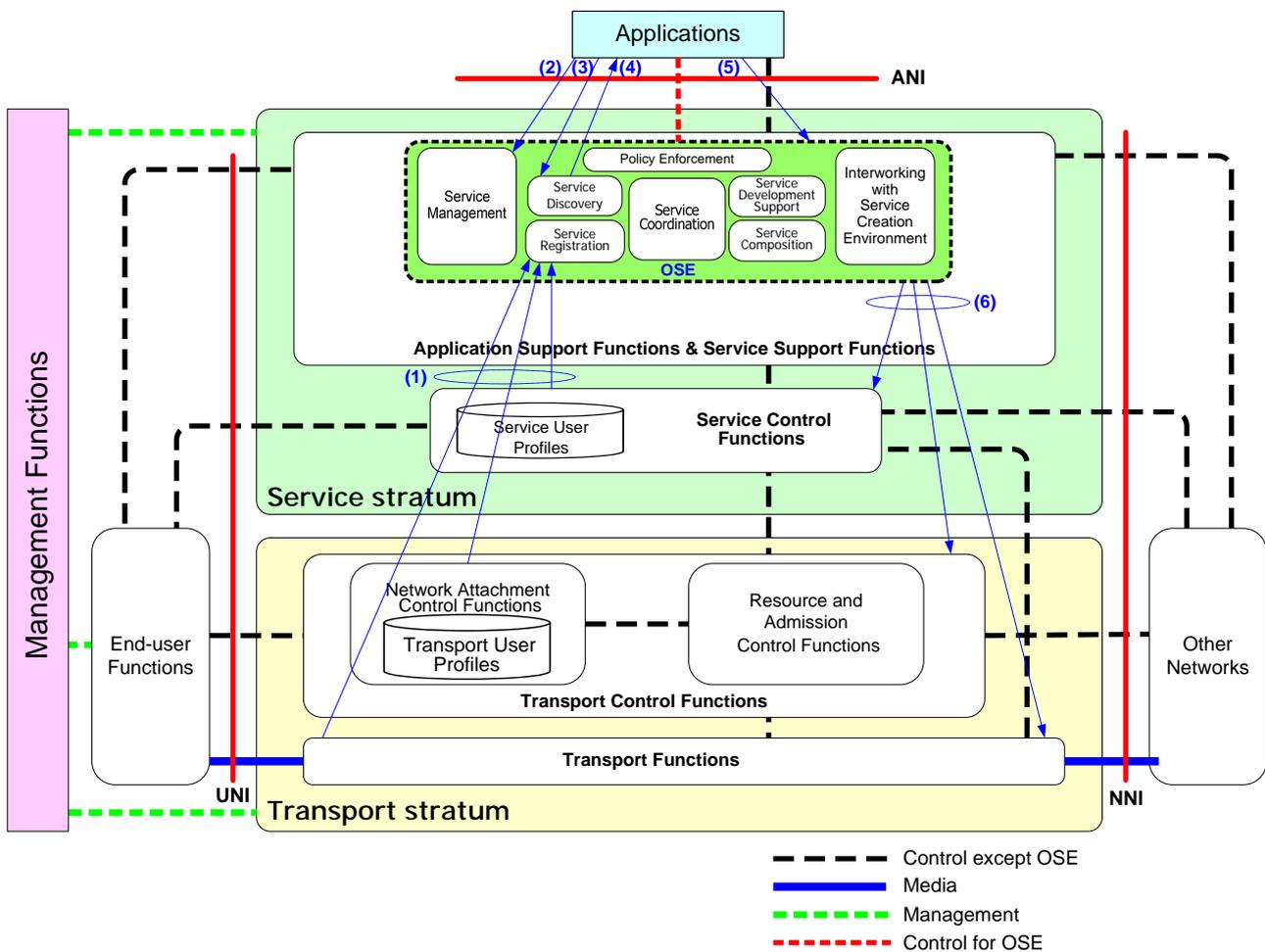


Figure I-1 – Interactions between applications, NGN services, and OSE capabilities

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 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 requests to service discovery the available services and selects one among them. Then it requests the associated interface information to 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

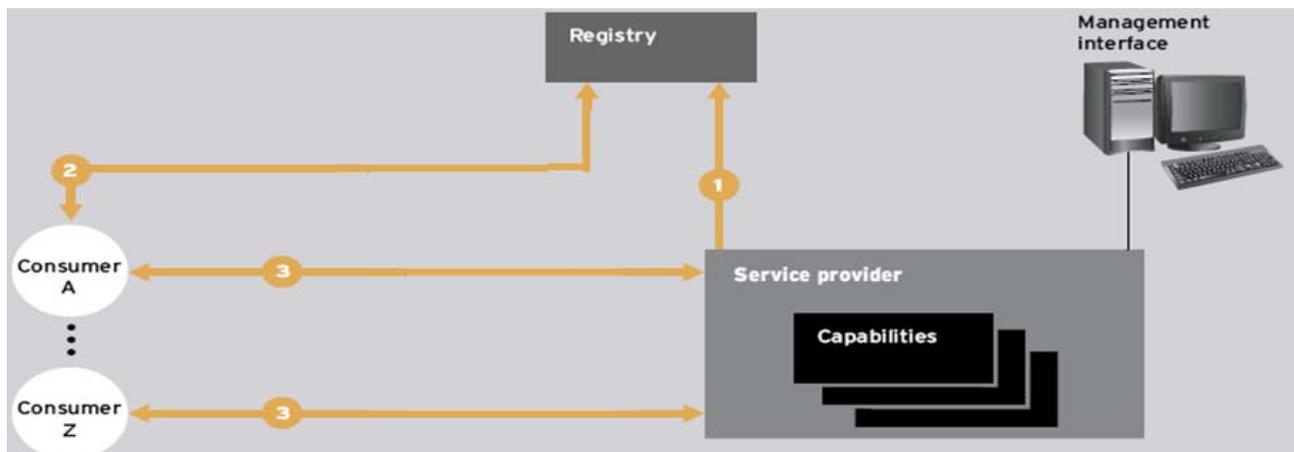


Figure I-2 – Interactions between service providers and service consumers

NGN open service environment comprises several major capabilities that allow potential services 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 rearrange 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 making use of the service discovery to discover the exact service component based on service request and then invoking 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 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 (see Appendix II for definition), new higher-level services can be created from simpler ones through choreography (see Appendix II for definition) using the OSE service composition. 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, say, 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, it could be converted to Business Process Execution Language (BPEL); and in the case of choreography, it could be converted to Choreography Description Language (CDL).

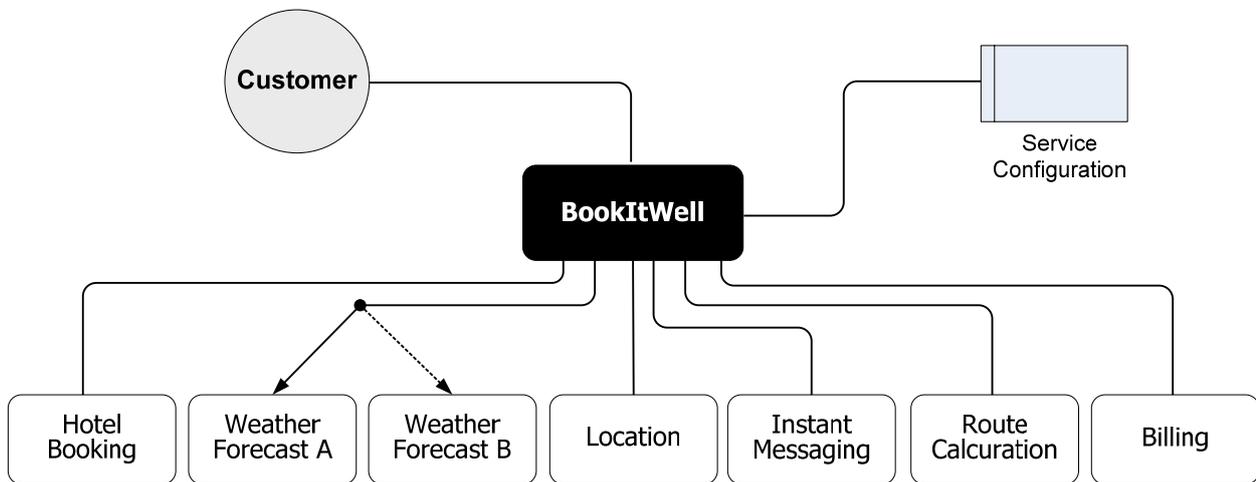


Figure I-3 – BookItWell service example

I.4 IPTV new customer application

This sub-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).

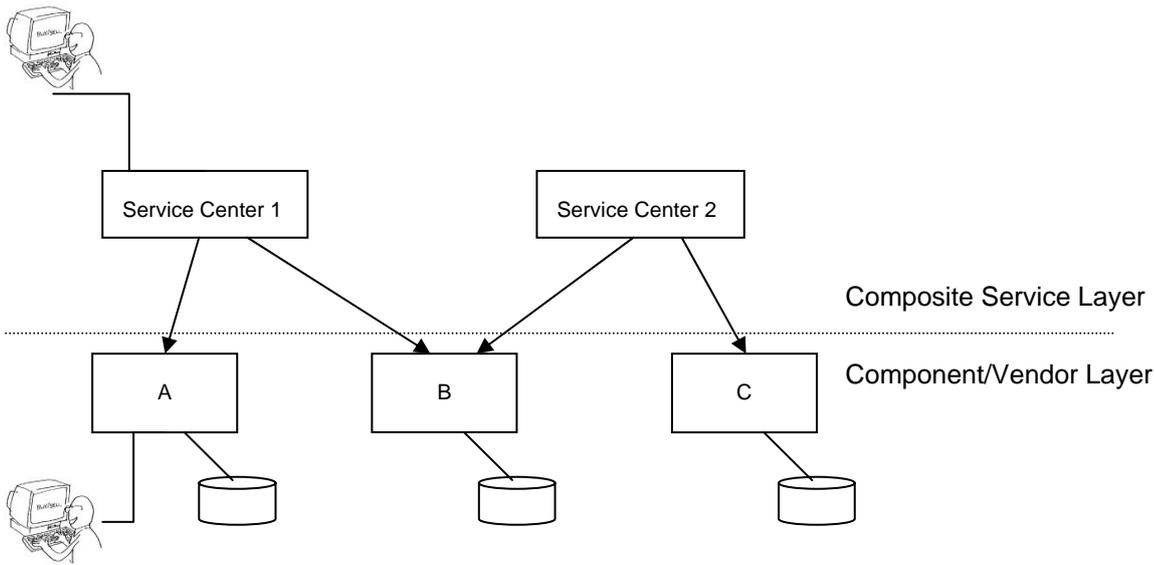


Figure I-4 – Telco service layered architecture

In the architecture shown above, let's 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, setup vendor-level IPTV services and setup financial information services.

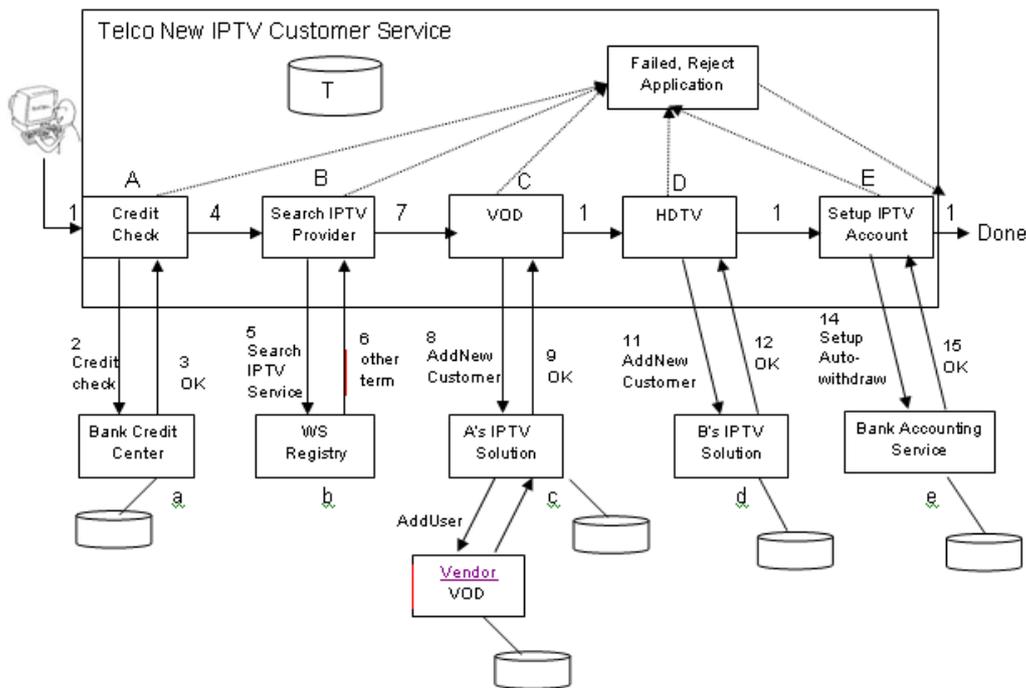


Figure I-5 – IPTV use case

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 center to see if this customer has a good credit standing.
3. The credit center 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 the Company A’s IPTV service provider to add the new customer and to enable the new service for this customer.
9. The 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 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 setup 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 setup 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 not become aware of these underlying changes and when services are properly implemented these changes should be transparent from a customer’s perspective.

Here are a couple of 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. Customer would like to choose a different package (i.e. Network VOD) via same Telco service provider. Telco service will need to search a fitting service provider and setup new service, and in the mean time 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)

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 SOA environment are made available as independent services that can be accessed without 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-Eric 2005] of SOA is a style of software design that guides all aspects of creating and using business services. A 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 encourages the use of standardized interfaces to make NGN capabilities accessible and easy to be used by third parties provider and applications developers.

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

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 defs].

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 defs].

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 API. They represent an abstraction and simplification of the Parlay 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 Services APIs are provided by the Parlay Web Services gateway which is an intermediary function between the Parlay application server and the Parlay/OSA gateway. Parlay X Web Service is a high level specification used by the Web Service application and it also specifies application friendly APIs.

OSA/Parlay [b-ETSI ES 203 915-3] information related to OSE capabilities is in sub-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 Environment provide standardized components to create an environment in which services may be developed and deployed.

OMA Service Environment [b-OMA OSE 2006] information related to OSE capabilities is in sub-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 sub-clause III.7.

III.4 World Wide Web Consortium (W3C)



Figure III-1 – The W3C technology stack

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

W3C is developing the WS-CDL (Web Services Choreography Description Language) that is targeted for composing interoperable, peer-to-peer collaborations between any type of participant regardless of the supporting platform or programming model used by the implementation of the hosting environment.

W3C [b-W3C] information related to OSE capabilities is in sub-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 Modeling 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 sub-clause III.7.

III.6 TMF: The TeleManagement Forum (TMF)

The TMF is an international membership organization of communications service providers and suppliers to the communications industry. While operations support system (OSS) is generally

dominated by proprietary and custom technologies, the TMF is focused on standards and frameworks for OSS.

The newest stage in OSS architecture work has come with the TMF's NGOSS programme establishing a set of principles that OSS integration should adopt, along with a set of models that provide standardised 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.

TMF [b-TMF] information related to OSE capabilities is in sub-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 2006]	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	None

NGN capabilities	OSA/Parlay [b-ETSI ES 203 915-3]	OMA OSE [b-OMA OSE 2006]	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 Modeling Language WS-Eventing	BPRI: Business Process Run time Interface Description: looking at runtime system, monitoring and measuring its and evaluating these measurements against what the expectations RAS: to publish the services	Service Delivery Framework (Work In Progress) a framework that supports and integrates all functions required for the lifecycle of a service delivered to Customer, 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 2006]	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	UPMS, BPMN, BPDM	None
Service Development Support		XDM, OSPE (OMA Service Provider Environment)		Service Modeling Language	<ul style="list-style-type: none"> - UPMS, - BPMN, - BPDM Existing Standards <ul style="list-style-type: none"> - 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)

NGN capabilities	OSA/Parlay [b-ETSI ES 203 915-3]	OMA OSE [b-OMA OSE 2006]	OASIS [b-OASIS]	W3C [b-W3C]	OMG [b-OMG]	TMF[b-TMF]
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	None
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

NGN capabilities	OSA/Parlay [b-ETSI ES 203 915-3]	OMA OSE [b-OMA OSE 2006]	OASIS [b-OASIS]	W3C [b-W3C]	OMG [b-OMG]	TMF[b-TMF]
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		None	None

Bibliography

- [b-ITU-T I.312] CCITT Recommendation Y.312 (1992), *Principles of intelligent network architecture*
- [b-Eric 2005] Eric Newcommer and Greg Lomow, “*Understanding SOA with Web Services*”, Addison-Wesley, 2005
- [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)*
- [b-OASIS] OASIS Web Site, <http://www.oasis-open.org/home/index.php>
- [b-OASIS SOA-RM] OASIS Reference Model for Service Oriented Architecture 1.0 (2006), OASIS Standard, 12 October 2006, <http://www.oasis-open.org/specs/index.php#soa-rmv1.0>
- [b-OMA] OMA Web Site, <http://www.openmobilealliance.org/>
- [b-OMA OSE 2006] OMA, Approved Version 1.0.3 (2006), *OMA Service Environment*
- [b-OMG] OMG Web Site, <http://www.omg.org/>
- [b-TMF] TM Forum Web Site, <http://www.tmforum.org/>
- [b-W3C] W3C Web Site, <http://www.w3.org/>
- [b-W3C defs] W3C Working Group Note (2004), *Web Services Glossary*, <http://www.w3.org/TR/ws-gloss>
-