Section 1

Concepts

1

IMS Service, Models, and Concepts

Emmanuel Bertin and Noël Crespi

CONTENTS

Introduction

NGN (next-generation network) is a concept that has been introduced to take into account the new situation and changes in the telecommunications fields. This new situation is characterized by a number of aspects: the deregulation of markets, the new demand from users for innovative services to meet their needs, and the explosion of digital traffic (increase of Internet usage). The introduction of NGN comprises economic and technical aspects. Economically, it allows increasing productivity by creating new usage [1] based on user preferences and related to voice and data services (e.g., voice over IP, instant messaging, presence, streaming, and push to talk). It also permits reducing costs for infrastructure maintenance, with only one type of transport network instead of specific ones for each access network. Technically, NGN makes the network architecture flexible in order to define and introduce new services easily.

The cornerstone of the service architecture for next-generation networks is the IMS (IP multimedia subsystem) architecture, standardized by 3GPP (3rd Generation Partnership Project). The IMS offers telecom operators the possibility to build an open IP-based service infrastructure that will enable easy deployment of new, rich multimedia communication services mixing telecom and data services.

The conception of IMS services is a key challenge for the telecom market. IMS services are fundamentally tailored to user preferences, rely seamlessly on multiple access networks, and bundle multiple service features (e.g., voice/video connectivity, community tools, presence, conferencing, gaming, and TV broadcasting).

The architecture and technical aspects of the IMS architecture are well addressed by the standardization bodies. However, a clear model of what an IMS service is (and what it is not) is not proposed by these bodies. The objective of this chapter is to detail the concepts behind IMS services and to propose a way to link IMS service, service building blocks, and technical functions.

This chapter is divided into three sections. In the first section, we present a survey of IMS services, starting by briefly introducing NGN architecture and then describing IMS service architecture and the OMA (Open Mobile Alliance) achievements. In the second section, we present how IMS services can be linked with service building blocks and with technical functions. In the third section, we illustrate the previous section with the case study of the push-to-talk over cellular service (PoC), specified by the OMA.

The Foundations of IMS Services

From IN to NGN

The concept of intelligent networks (INs) developed in the 1980s was a precursor of the NGN. The principle of INs is to separate clearly the switching functions from the service data and logic located in an external entity: the service control point (SCP). A new functional entity is added to the TDM switch, the service switching point (SSP), which interfaces between the service logic and the switch itself. An interface based on the intelligent network application part (INAP) protocol family is introduced between the SSP and the SCP. The services are no longer developed in the TDM switch—as with the concept of global system for mobile communications (GSM) and inte-

AU: pls sp. out for 1st use

grated services digital network (ISDN) supplementary services—but rather are implemented in the SCP. The INAP and associated procedures allow the SCP to control and monitor the switch.

The intelligent network introduced the concept of a service independent building block (SIB) for reusable service functions. A service could thus be thought of as a composition of various SIBs. But this goal was not fully achieved because of a lack of independence with INAP protocol, a lack of software reusability, and a lack of openness by manufacturers and operators. As a consequence, INs deployed today rely on a monolithic architecture and service platforms do not offer flexible services. In addition, as the service logic is executed in external entities, triggering multiple services for one call requires having service interaction management mechanisms. This issue, known as feature interaction, is one of the most complex problems encountered in IN and considerable work has been done on it. However, this work cannot be directly applied to the NGN because of the service and architectural differences between IN and NGN.

The promise of the NGN, as defined in the late 1990s, was to offset these shortcomings by moving from a vertical approach (where access, control, and services are closely tied) to a horizontal approach (where each layer provides reusable elements to other layers). Specification work is ongoing at the International Telecommunication Union (ITU)-T (as described in Knightson, Morita, and Towle [2]) to formalize the separation (e.g., through standard protocols or application programming interfaces [APIs]) between

- the transport stratum that is composed of transfer functions from various access networks (UMTS terrestrial radio access network [UTRAN], wireless local area network [WLAN], xDSL) and from the core networks, control functions for these transfer functions (e.g., network attachment control or resource and admission control), the transport user profiles (e.g., to store the data linked to network attachment), and the media handling functions (e.g., for playing announcements or for transcoding); and
- the service stratum composed of access-independent service control functions (e.g., session establishment control or service triggering control), application functions, and service user profiles. Application functions should be independent from the service control functions and should offer flexibility (e.g., by using open software mechanisms) to answer user needs.

This NGN architecture with two strata is defined at the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) (Figure 1.1). The NGN architecture may also be represented with three layers instead of two strata (this is, for instance, the case at the European Telecommunications Standards Institute [ETSI]). In this case, service control functions and transport control functions are grouped into a control layer. The

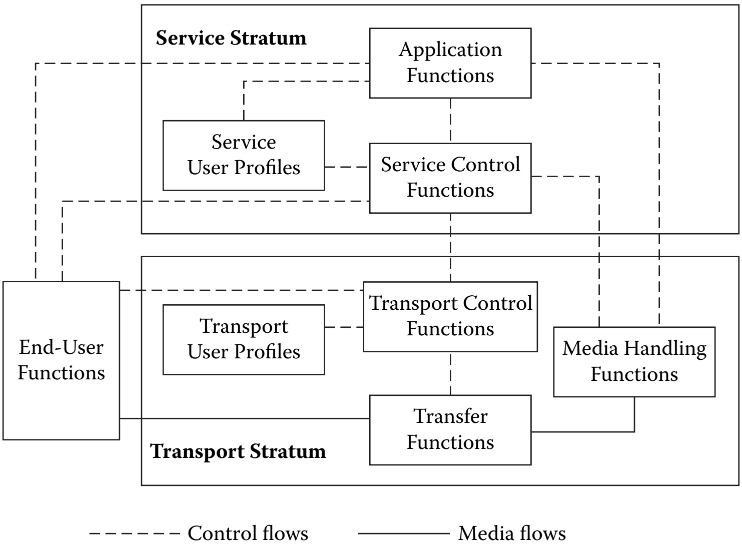

FIGURE 1.1
NGN technical architecture [2].

separation thus involves a transfer layer (with transfer functions), a control layer (with transport control functions and service control functions), and an application layer (with application functions).

We can draw a parallel between IN and NGN architectures: The service control function (usually implemented with a session initiation protocol [SIP] proxy) is the NGN counterpart of the TDM switch/SSF (service selection function and the application function (for example, implemented with a SIP application server) is the NGN counterpart of the service control function (SCF). In both architectures, the triggering criteria have been defined in order not to invoke services systematically but only when required. However, there is a key difference between those architectures regarding the triggering mechanisms. In IN, the SCF controls the SSF using INAP, which is independent of the call control protocols. In the NGN architectures, the application function is inserted in the signaling path; therefore, all SIP signaling requests and responses can be intercepted by the entity controlling the services. Indeed, the IN concept of "point of control" (i.e., an entity that can control the SSP and modify the signaling at any time) does not exist in the NGN context. This concept is replaced by the notion of application function present in the signaling path, which can modify SIP messages to execute a service logic. The consequence of this fundamental difference in signaling and architecture is that mechanisms defined in IN for feature interaction are mostly not applicable for SIP.

From NGN to IMS

The IMS architecture is a realization of NGN principles, relying on the SIP protocol for the session control. The IMS specifications [3] define the whole multimedia session control architecture on top of the universal mode telecommunications system (UMTS) packet-switched domain. With IMS, operators provide both reliable session control and better integrated services. Because IMS is solving architectural issues for SIP deployments (as detailed in Bertin, Bury, and Lesieur [4]), it is now seen as a guideline for all SIP deployment using the client/server paradigm. While the IETF (Internet Engineering Task Force) has standardized the SIP protocol but not the associated architectures [5], the 3GPP has defined with precision the architectures and the procedures to ensure roaming, scalability, security, and reliability. Moreover, the IMS specifications are not intrinsically linked to mobile networks [6]. IMS was, for the most part, conceived independently from the UMTS packet-switched domain and can be adapted to other types of access networks. 3GPP has specified the interface between IMS and WLAN access networks (IMS release 6) [7]. The ETSI TISPAN (Telecommunications and Internet Converged Services and Protocols for Advanced Networking) project specifies the adaptations controlling xDSL access networks with IMS [8]. In addition to IMS, TISPAN is also defining other subsystems such as public switched telephone network (PSTN)/ISDN emulation for PSTN replacement (which will be needed in Europe between 2008 and 2012).

The major elements related to service architecture are the following:

- S-CSCFs (serving call state control functions) implement service control functions (session control and service triggering).
- HSS (home subscriber server) is the central service and network database. It implements the service user profiles (as well as the transport user profiles).
- ASs (application servers) implement the application functions, providing session-related services to users. The ASs offer APIs like OSA/Parlay or SIP servlet for application execution.

Concerning user identity, the user is represented in IMS by several identifiers. Public identities are routable addresses that can be communicated to the contacts of the user and can be used to reach this user (e.g., sip:alice@provider.com or tel:+33123456789). Private identities belong to the IMS operator and are stored in the SIM (subscriber identity module) card. The same user may have several private user identities and several public user identities, but only one private identity is stored per SIM card (Figure 1.2).

Concerning service triggering, IMS provides an application triggering architecture based on filter criteria and service points triggers (SPTs) [9]. Initial filtering criteria (iFC) allow the S-CSCF to decide which services should be invoked during a SIP session or transaction and in which order they should apply. The SPTs are the points in the SIP signaling on which filter

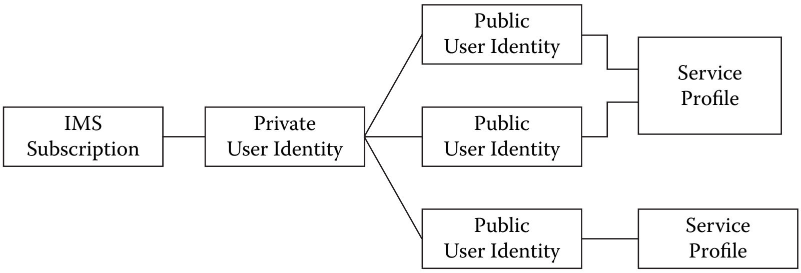

FIGURE 1.2
IMS user identities in IMS release 5 [3].

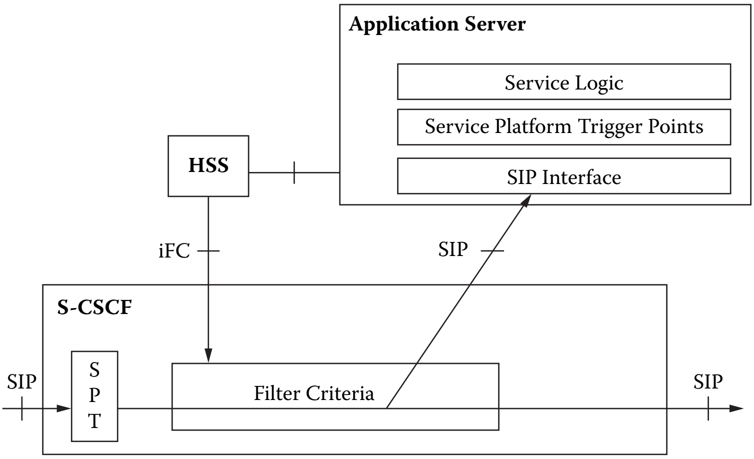

FIGURE 1.3
Application server triggering architecture [9].

criteria can be set. The filter criteria are distributed among the S-CSCF, HSS, and IMS application server, as shown in the Figure 1.3.

iFCs are stored in the HSS as a part of the service profile. They are downloaded to the S-CSCF upon user registration or upon a terminating initial request for an unregistered user. They are active during the registration lifetime or until the service profile is changed. Filter criteria should contain the following information, structured in an XML format:

- the address of the AS to be contacted;
- the priority of the filter criteria providing the sequence in which the criteria will be applied;

- the SPTs, which may contain the following information: SIP method, presence or absence of any header, content of any header, session description information, etc.;
- default handling if the AS is not reachable; and
- optional service information, added to the message body before it is sent to the AS.

During the registration phase, an S-CSCF is assigned to control user services. The service profile (containing iFCs) of the user is downloaded from the HSS to the S-CSCF. When the S-CSCF receives a SIP request matching the iFC, it invokes the associated service by forwarding this SIP request to the AS indicated in the iFC. iFCs are only applied to initial SIP requests (i.e., the requests initiating a SIP session or transaction: INVITE, SUBSCRIBE, REGISTER, OPTION, etc.); consequently, the service invocation can be done only statically in the SIP session or transaction initiation phase.

A user may subscribe to several services, and as a consequence several iFCs may be present in the service profile. When the S-CSCF receives an initial SIP request, it checks whether it matches the iFC that has the highest priority for this user. If it does not match, the S-CSCF checks the next iFC, in the predefined priority order. If it matches, the S-CSCF forwards the request to the indicated AS. This AS executes the service logic, eventually modifies the request, and sends it back to the S-CSCF. The S-CSCF performs the same processing with the next unexecuted iFC. The S-CSCF continues this process until all the iFCs are checked. The AS may also suppress the information required to trigger the iFC (e.g., replacement of public identity by a globally routable user agent [UA] uniform resource identifier [URI]) or locally end the request as a part of the service logic (e.g., a prepaid account without remaining credit). These mechanisms will be used to build future communication services with the IMS.

3GPP had specified a SIP AS called service capability interaction manager (SCIM) for managing the interactions between application servers, but neither "the service invocation functionalities over ISC" nor "the service interaction management functionalities of SCIM" are specified in the standards [14]. These points are detailed in Chapter 14, "Service Orchestration in IMS."

IMS Service Capabilities and OMA Enablers

The business purpose of the IMS is to enable the building of innovative services in a flexible way. IMS services will include multiple service features like chat, instant messaging, voice, video, presence, address book, and TV broadcasting [10,11]. If all these features are deployed in an uncoordinated way by a service provider, the user will have to handle the interaction between the services (e.g., by entering the same personal preferences several times). In addition, advanced services that combine many service features (like routing voice calls according to the originating community and the availability state) are not possible if there is no coordination between features. The answer to improving user experience is to build a coherent service environment by standardizing the applications functions.

Standardization of application functions is today mainly driven by ITU-T, 3GPP, and OMA. Telecom and IT companies regroup within OMA to specify interoperable advanced mobility services. OMA was created in June 2002 as a combination of the WAP forum, the SyncML Initiative, the MMS Interoperability Group, the Wireless Village Initiative, the Mobile Wireless Internet Forum, and the Mobile Games Interoperability Forum. The goal of ITU-T, 3GPP, and OMA is not to standardize complete services but rather to standardize functional service building blocks that are reusable at runtime by various services, as defined in Bertin, Bury, and Lesieur [13]. This approach enables the building of innovative and evolving services mostly independently of network considerations. These service building blocks provide key capabilities to ensure interoperability of devices, operators, and service providers. As seen before, ITU-T and 3GPP are standardizing the mechanisms that trigger these building blocks, either separately or in a coordinated way, including the management of interactions between these capabilities, as shown in Gouya, Crespi, and Bertin [14]. These service building blocks are called service capabilities at 3GPP, service support capabilities at ITU-T, and service enablers at OMA. Service support capabilities studied at ITU-T [15] typically include presence, location, group management, message handling, broadcast/multicast, push and session handling or device management. Service enablers at OMA [16] include, for example, data synchronization, device management, digital rights management, downloading, e-mail notification, instant messaging, presence and mobile location or multimedia messaging. Service capabilities defined at 3GPP typically include presence [17] and messaging [18] or conferencing [19].

The OMA specifications for service enablers are the most advanced and complete. According to the OMA,

- "[An enabler is defined as] a technology intended for use in the development, deployment or operation of a service; defined in a specification, or group of specifications, published as a package by OMA" [20].
- "An enabler should specify one or more public interfaces. Examples of OMA enablers include location or device management" [16].

These definitions highlight the normative character of an enabler. A component or a technology is an enabler because it has been defined as an enabler. Moreover, when individual enablers are defined independently, each enabler has to define all functions required to fulfill its requirements. This implies several issues for the service provider—especially the difficulty of providing user-centric services: "Integration and deployment of services is complicated and expensive; high implementation efforts for applications wanting to use several capabilities; there is no common integration of the different services from the point of view of the end user (e.g., no common group management or user profile across multiple services)" [16]. An OMA enabler should thus contain only intrinsic functions that can interact with other functions from the service architecture and/or from underlying network architecture. Intrinsic functions are defined as "those functions that are essential in fulfilling the intended task of the specified enabler. For example, the position calculation function is intrinsic to secure user plane location; authentication is intrinsic to single sign on; encryption is an intrinsic function of digital rights management" [16].

This separation into intrinsic and nonintrinsic functions is a way of assuring that various enablers will not include the same function (e.g., authentication function in each enabler). As specified in reference 16, "any requirements or features that are not intrinsic to an enabler should not be specified within the enabler's specification. An enabler's specification should only specify the intrinsic functionality required to fulfill its actual function." This specification of service functions with enablers that are responsible only for their intrinsic functions enhances the ability of service providers to offer a consistent user experience (i.e., reuse of user information, service continuity, etc.). However, the separation into intrinsic and nonintrinsic functions is not obvious but remains subjective, as recognized in reference 16 ("the classification of intrinsic and non-intrinsic is subjective and needs to be done on a per enabler basis"). This implies again that the definition of enablers should result from a normative process.

The OMA has specified the OMA service environment (OSE) [16] that provides a common architecture for the integration of enablers and service creation. As shown in Figure 1.4, the OSE architecture consists of enablers that run on an execution environment, and are accessible to applications and other enablers through a policy enforcer.

Enablers are intended for use in the development, deployment, or operation of a service. They provide their intrinsic functionality through one or more public interfaces called I0 interfaces, and may use underlying network resources through I2 interfaces (such as IMS interfaces) The execution environment logically encompasses various functions such as process monitoring, software life cycle management, system support (e.g., thread management, load balancing, and caching), operation, management, and administration. The interface between the execution environment and enablers is called I1 interfaces. The policy enforcer provides a policy-based management mechanism to protect resources from unauthorized requests and to manage the use of these requests—for instance, through appropriate charging, logging, and enforcement of user privacy or preferences. The policy enforcer function allows the domain owner to extract and separate policy rules from architectural elements. This element exposes I0 + P interfaces to applications and enablers, where P is additional parameters that must be provided along with a request to an enabler's I0 interface, when the policies that are to be enforced

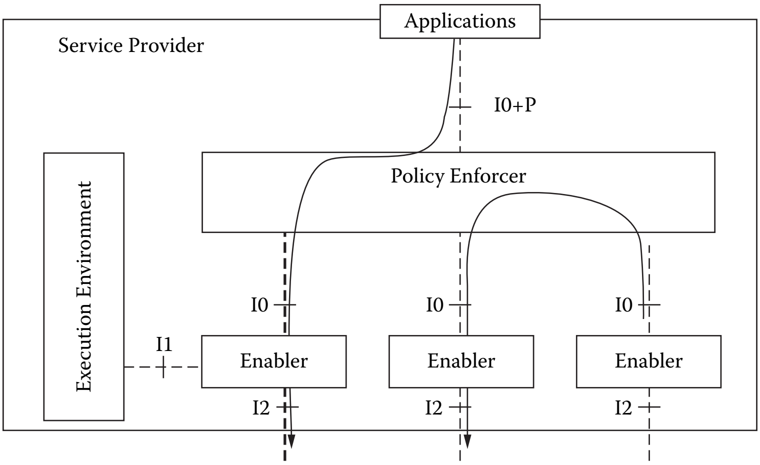

FIGURE 1.4
The OMA service environment architecture [16].

require additional parameters. Applications can be located inside or outside the service delivery environment.

IMS Service Model

IMS Brings New Types of Services

Traditionally, telecommunications services are divided into bearer services, teleservices, and supplementary services. “A bearer service is a type of telecommunication service that provides the capability for the transmission of signals between user-network interface” and “a teleservice is a type of service that provides the complete capability, including terminal equipment functions, for communication between users” and “supplementary service modifies or supplements a basic teleservice” [26]. Examples of basic teleservice are telephony, facsimile, or emergency calls.

These notions are still in use in some 3GPP or TISPAN standards, but can no longer be used by a service provider to design services. Indeed, the added value of IMS for service providers is the ability to build user-centric services that flexibly combine several features and enable the sharing of user information between these features to form a coherent service environment for the user [12]. As explained in the previous section, the OMA enabler or the 3GPP service capabilities are the necessary building blocks for such services. However, a model for IMS services, linking the services from users, enablers, and technical functions, is not defined in standards.

Transfer and control functions are extensively addressed by IMS and NGN studies. Application functions are partially addressed by the OMA concerning the service delivery aspects (with the OSE). The foreseen services for IMS will require a coherent integration of multiple loosely coupled features. The integration between these features should be considered not only at the technical level (i.e., the integration within a service delivery environment like OSE), but also at a service level (i.e., how the composition of various technical functions and enablers will provide a coherent service experience to the user). If the integration at the technical level is well addressed by the OMA and ETSI studies, the integration at the service level has not been investigated.

To answer these needs, we should describe the relationships between a service perceived by the user and the technical functions and enablers used to implement it.

The modeling approach is organized as follows:

- modeling the link between services that is seen by the users (e.g., a user is aware that his personal information is shared between his services);
- modeling the technical functions that are the foundation of IMS; technical functions are those carried out by the systems (e.g., service platforms, terminals, etc.) controlled by the service providers; and
- modeling IMS service architecture based on service enablers. Service enablers are designed for the reuse of the user information between services and for an easy integration of new services. As seen before, service enablers contain and wrap technical functions (intrinsic functions). We propose to characterize an enabler by the information it handles and by the technical functions it wraps. For instance, only one service enabler can produce the presence information and can wrap the technical functions linked to presence or only one service enabler can produce the location information and can wrap the technical functions linked to location.

The Link between Services Seen by the User

The first step is to define clearly what a service is. There is a lot of research on the notion of service—not that much in the IT area, but rather in the economic and business sciences, as surveyed in Ben Yahia et al. [21]. In a generic way, a service can be defined as any business actions or business activities that have a value-added result for a user (a person or a system). This action or activity is offered by a service provider (another person, entity, or system), which profits from providing this action [22,23].

In the telecommunications field, a telecom service is defined by 3GPP as “a component of the portfolio of choices offered by service providers to a user, functionality offered to a user” [24].

The focus area of this study is service usage; hence, we concentrate on the user while the customer is outside the scope of IMS services. The customer is a person or organization that purchases products and services [25]; the user is the person (or system) that uses the service and can be different from the customer. For example, in a family, the customer may be one of the parents, and a child may be the user of the purchased service. The customer usually assigns rights to users to use the services he has obtained. The customer can be a user himself. Although the user is typically a person, it may also be another actor (e.g., another service provider).

Relying on the preceding service definition, we propose a definition for IMS services as follows:

> Activities that take place in interactions between a user (i.e., IMS user) and systems controlled by service providers (e.g., IMS user equipment, IMS platforms). These activities have a value-added result for the user; and the service providers profit from providing these activities.

In this definition we highlight two parties: the user and the systems controlled by the service providers.

From a user perspective, the purpose of IMS services is to establish a communication session between users that is adapted to user preferences and context. The session manipulated by IMS services may be voice sessions, but can also be video sessions, instant messaging sessions, or collaboration sessions. The term *session* here means only an interactive exchange between two or more persons in order to communicate. From a user perspective, an IMS service is linked to his identity and not to his access device because the user may access the same services from several IMS devices.

When using his IMS services, the user is aware that applications within his user equipment or within service platforms are sharing and reusing his personal information such as his presence information, availability rules, personal profile, contact list, or location information. A given service will be responsible for the creation and the modification of each type of information (e.g., presence service for presence information, location service for location information, etc.). An IMS service can thus consult a user's personal information (according to privacy policies) and may be responsible for defined user information.

Figure 1.5 proposes relationships of an IMS service, an IMS public user identity, and the user's personal information. The terms of IMS service in this figure do not name a service in a general way (e.g., presence service), but name the service instance of one specific user (e.g., Bob's presence service).

Technical Functions

From a service provider technical perspective, a service is implemented with technical functions. Technical functions are the functions carried out by systems controlled by the service providers (e.g., service platforms, terminals, etc.). As seen before, the IMS service architecture may be divided into several technical functions. The first division is among service stratum functions, transport stratum functions, and end-user functions. As we are not dealing

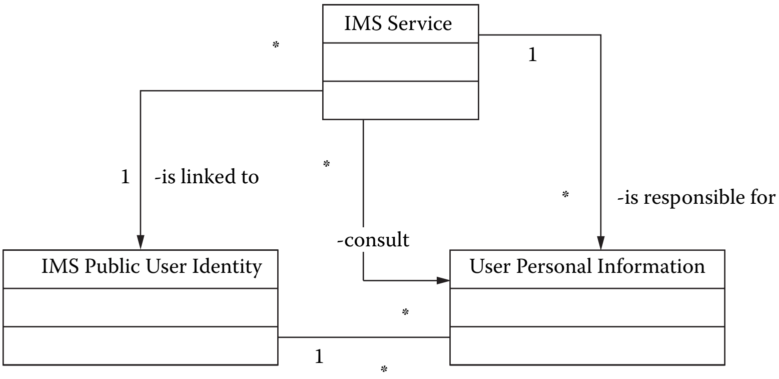

FIGURE 1.5
Links seen by the user.

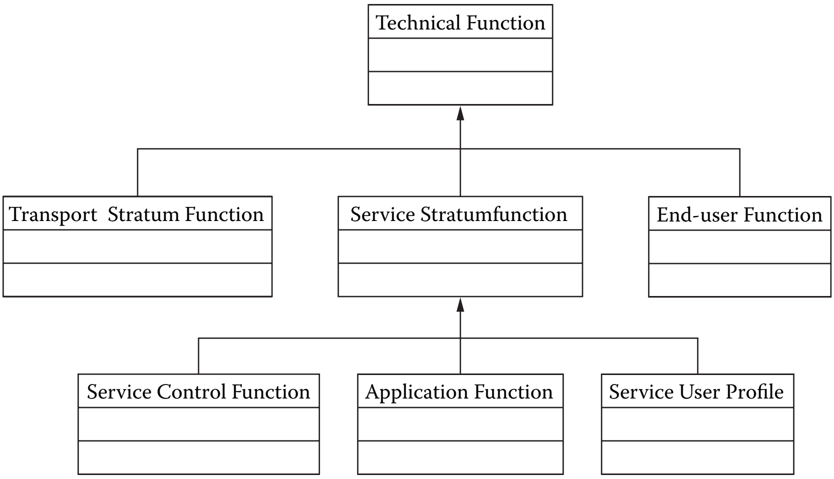

FIGURE 1.6
IMS technical functions.

here with networking issues, we will focus only on the service stratum. As seen in the first section, this service stratum is divided among service control functions, service user profiles, and application functions [2]. In addition, end-user functions have to be considered. They are not part of the service stratum but are closely related for the delivery of the services through the user interface.

Figure 1.6 classifies the IMS (or NGN) technical functions, according to the NGN standards presented in the first section. The service stratum functions are a particular type of technical function. A service stratum function may be:

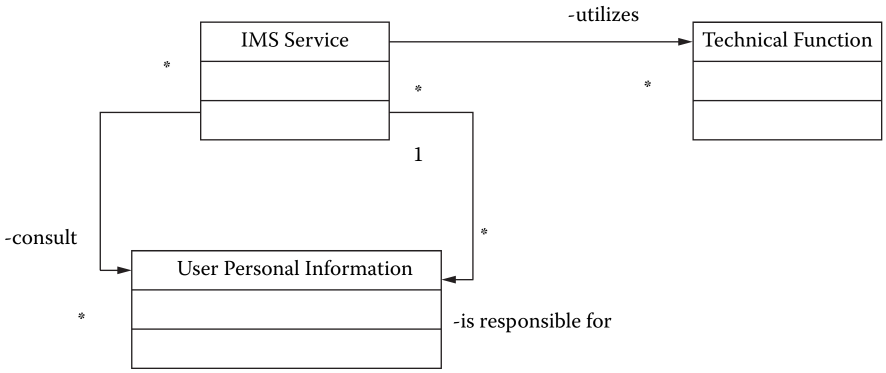

FIGURE 1.7
IMS service.

- a service control function that handles common control functions like session establishment control or service triggering control;
- an application function that contains the service logic and the manipulation rules for session establishments (e.g., transfers, callback, reachability, call log, etc.);
- a service user profile that stores the information on user identities and on service triggering; and
- an end-user function that includes not only the connection to the IMS (using SIP and bearer protocols) but also the service interface part that resides in the client device. This interface performs the transformation of the technical messages from the application functions into something usable by the user (and vice versa) and thus provides the end user with the ability to initiate and participate in a session. For example, an interface for presence will transform the presence protocols messages into a user interface displaying the presence of the user's contacts.

Relationship between Service and Technical Function

An IMS service is the conjunction between user personal information and technical functions. To illustrate this in Figure 1.7, we can consider the example of an IMS presence service. The presence service is seen by the user as the notification of presence information between a consumer of presence information and sources of presence information, where the presence information is a set of attributes characterizing current properties of the sources (such as status or communication address) [17]. The presence service is performed with technical functions such as end-user presence clients (a presence source client and a presence watcher client), service control mechanisms to route presence messages (the SIP SUBSCRIBE, PUBLISH, and NOTIFY messages), and presence application servers (to process the presence state from the presence sources and to store and send it to the watchers that have subscribed to this presence event).

The services are directly responsible for the user's personal information and are utilizing the technical functions directly. As mentioned, this may lead to building silo architecture, where each service relies on its own technical functions. Service enablers (or service support capabilities or service capabilities) are designed to address this issue by focusing only on their intrinsic functions. This means that there should be no overlap between the service enablers, both from the user perspective and from the technical functions perspective.

No overlap from the user perspective implies that different service enablers should not be responsible for the same type of user's personal information. For example, only one service enabler can produce the presence information or only one service enabler can produce the location information.

No overlap from the technical functions perspective implies that the different service enablers should not use the same IMS functions in an incoherent way. For example, only the presence service enabler can process the presence messages and store the presence state or only the location service enabler can process and aggregate user location from various location sources.

In IMS service architecture, the IMS services have to rely as much as possible on IMS service enablers. These IMS service enablers wrap a set of technical functions and provide a consistent service interface to IMS services. An IMS service might also use some technical functions directly (e.g., an application server dedicated to a specific service). In addition, only IMS service enablers should be responsible for the user's personal information (Figure 1.8).

AU: inserted citation for fig 8 here—OK?

Example of the Push-to-Talk over Cellular

In order to illustrate this model, we apply it here to the push-to-talk over cellular (PoC) described in the OMA release program and specifications [27]. The PoC service is a walkie-talkie type of service that allows rapid, short, and spontaneous communications. It is a half duplex voice service that allows person-to-person and person-to-group communications. This service is considered an early example of IMS application in the market. Because PoC is specified as both a service and an enabler, we show the distinction between the service perceived by the user and the functional service building blocks.

This illustrates the separation of concern from what is seen by the user, the service enabler, and the technical functions that implement these enablers.

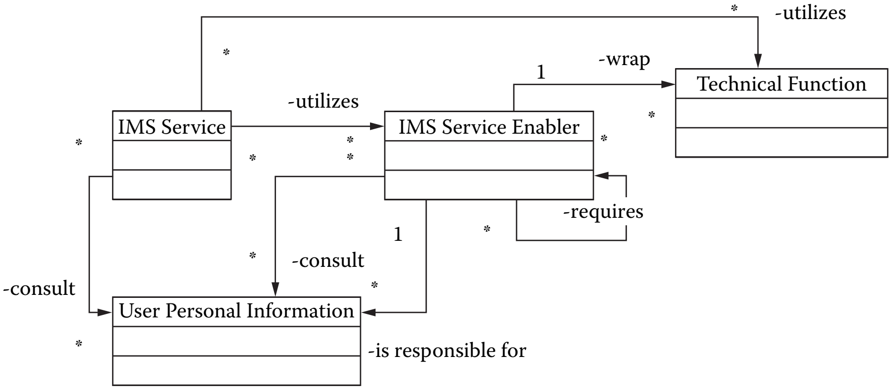

FIGURE 1.8
IMS services and service enablers.

This separation will benefit service providers for the whole service life cycle—especially service composition, service interaction, and service management.

PoC Service Seen from the User's Perspective

From a user perspective, a typical PoC session is as follows:

> The PoC user opens his contact list where presence features indicate whether contacts or groups of contacts are available or not. The user selects one or more contacts in his contact list, creates a PoC group with these contacts, starts the PoC service, and then talks simultaneously to all the contacts of his PoC group.

This basic session shows that the PoC service is based on the user identity, which is necessary to access the contact list and invite other PoC users to participate in a session. Besides identity, from a user perspective, the PoC service also uses:

- presence information to be aware of contact availability and reachability;
- contact lists to create groups for PoC sessions; and
- user profiles.

Figure 1.9 shows the PoC service as seen by user "Bob Smith." This view contains the information that the user owns and that is reused in the PoC service. His personal information could be reused as in another IMS service.

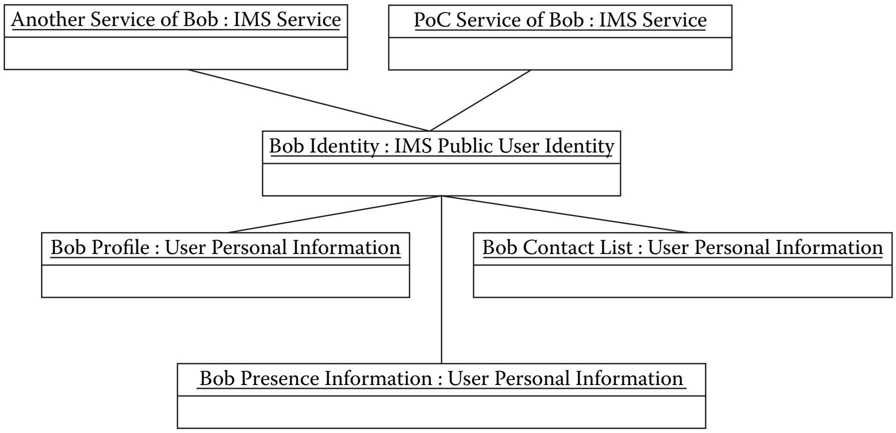
Another Service of Bob : IMS Service
PoC Service of Bob : IMS Service
Bob Identity : IMS Public User Identity
Bob Profile : User Personal Information
Bob Contact List : User Personal Information
Bob Presence Information : User Personal Information

FIGURE 1.9
PoC service as seen by "Bob Smith."

PoC Service and Service Enablers

As described in the OMA specifications, the PoC service requires several service enablers that perform specific actions and are responsible for specific information:

- push-to-talk over cellular enabler [27] that manages the service logic of the PoC service;
- XDM (XML document management) enabler [28] to handle the contact groups in particular;
- presence enabler [29];
- IMS enabler [30] to support the service; and
- device management enabler [31].

The dependencies between the PoC service and the service enablers and also between the service enablers are described in Figure 1.10 with dotted arrows. Each service enabler is responsible for some type of personal information.

Technical Functions for PoC Service

As mentioned before, each service enabler is implemented and carried out via a set of technical functions that are shown in Figure 1.11. In this section we split each enabler into its corresponding technical functions

The XML document management (XDM) enabler is implemented with an XDM client (XDMC), a shared XDM server (shared XDMS), and an aggregation proxy. The XDMC is an XCAP (XML configuration access protocol)

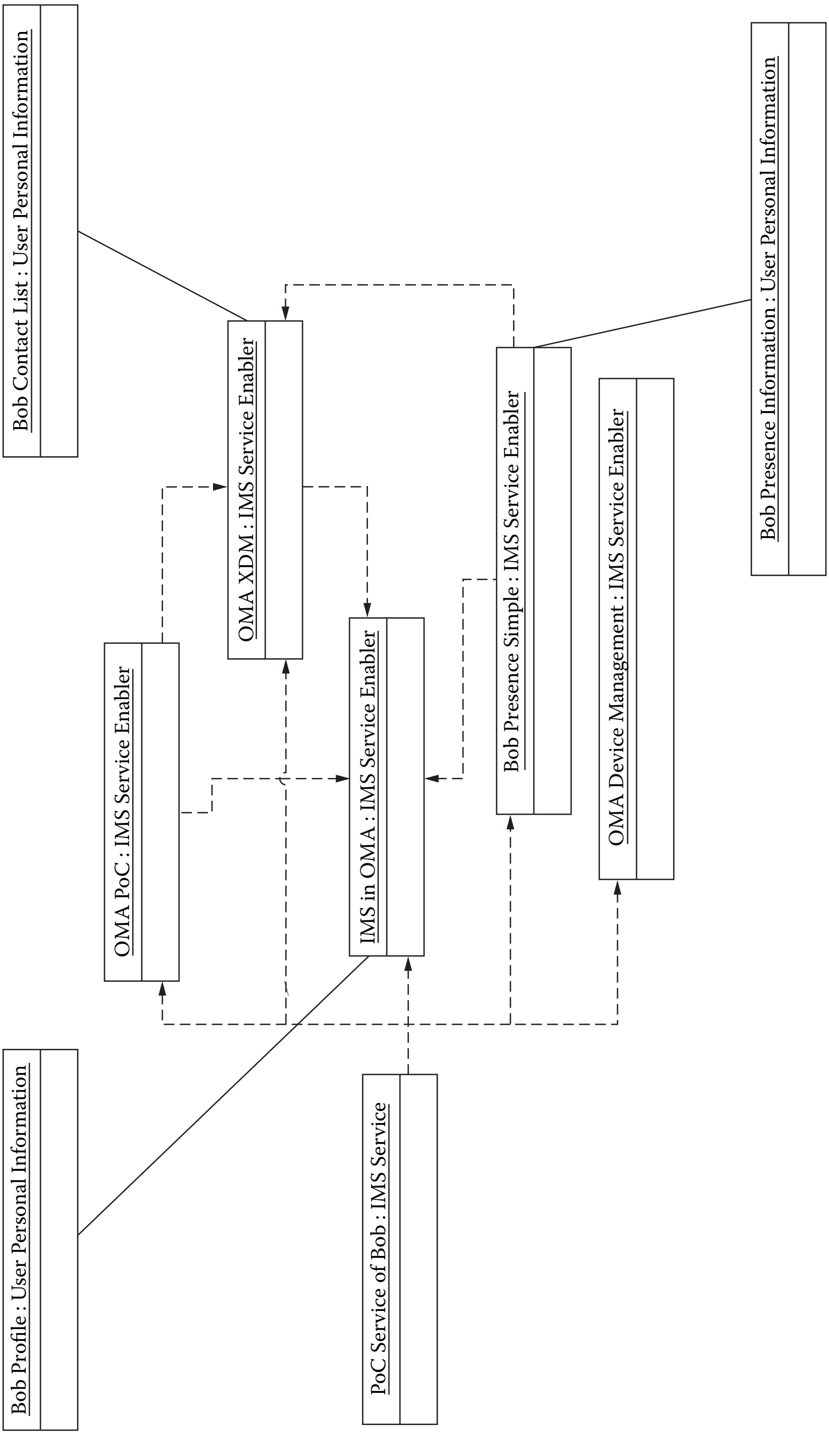
Bob Contact List : User Personal Information
Bob Profile : User Personal Information
OMA PoC : IMS Service Enabler
OMA XDM : IMS Service Enabler
IMS in OMA : IMS Service Enabler
PoC Service of Bob : IMS Service
Bob Presence Simple : IMS Service Enabler
OMA Device Management : IMS Service Enabler
Bob Presence Information : User Personal Information

FIGURE 1.10
Service enablers for PoC service.

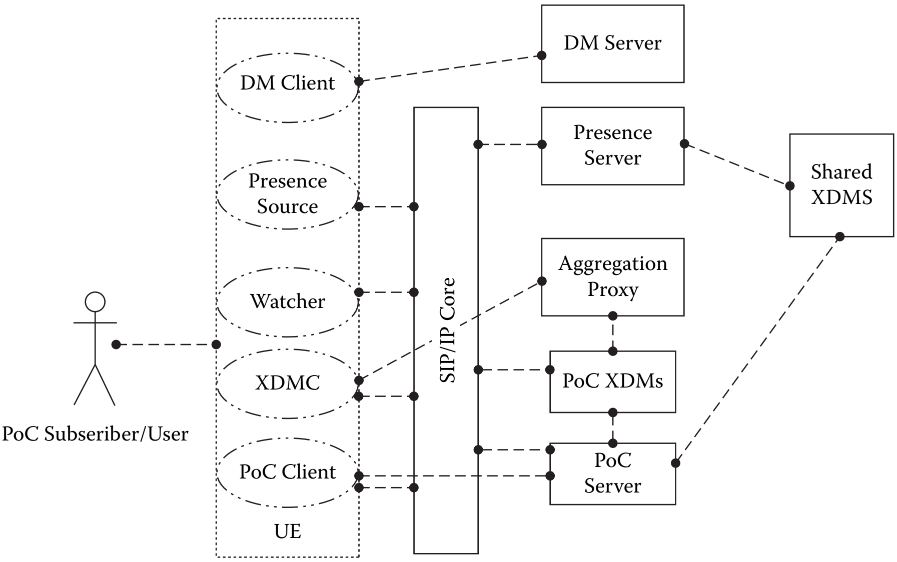

FIGURE 1.11
Technical functions of the PoC service (simplified).

client that gives access to XML documents stored in the network (e.g., PoC-specific documents in the PoC XDMS, contact lists in the shared XDMS, etc.). The aggregation proxy acts as the single contact point for the XDMC. It performs authentication of the XDMC and routes individual XCAP requests to the correct XDMS. The shared XDMS is an XCAP server that manages XML documents (e.g., contact lists) that are shared with other service enablers (e.g., presence).

The PoC enabler is implemented into a client part, a server part, and a PoC specific XDM server. The PoC client resides on the terminal and is used to access the PoC service. The PoC server implements the application logic for the PoC service. The PoC specific XDM server is an XCAP server, which manages XML documents that are specific to the PoC service (e.g., PoC groups).

The presence enabler is implemented on a presence server, a presence source, and a watcher. A presence server is an entity that accepts, stores, and distributes presence information about PoC clients. A presence source is an entity that provides (publishes) presence information and a watcher is an entity that is notified from presence information.

The IMS enabler includes a number of SIP proxies and SIP registrars. It performs functions such as authentication, authorization of PoC user, or maintaining of the registration state.

The device management enabler is implemented with a device management client that receives the initial parameters needed by the service provider for the PoC client and a device management server that initializes the entire configuration and updates necessary for the PoC client.

All technical functions described here belong to the service stratum. They are thus end-user functions, service control functions, or application functions. PoC client, XML document management client, presence source, watcher and device management client are end-user functions. IMS core is a service control function. PoC server, PoC XML document management server, aggregation proxy, shared XML document server, presence server, and device management server are application functions.

A Comprehensive View of IMS Services

Figure 1.12 is an example for the three enablers OMA XDM, IMS in OMA, and OMA presence simple. It defines the suitable dependencies of these three enablers and with the services that make use of these enablers. We take here the examples of the PoC service and of an instant messaging service. All the enablers used by these services are not represented in order to simplify the figure.

AU: should this be SIMPLE— SIP for instant messaging and presence leveraging extensions?

Conclusion

IMS services cannot be considered independently from the whole service environment of the user [32]. This environment includes at least features such as identity management, community management, availability management, or context management. This service environment should be able to integrate third-party service elements. The service value will reside in the quality of the interactions between all the service elements and in seamless accessibility in a user-centric way. A service control framework handling these interactions is therefore needed for the interactions between the operator services and for intermediation with other service providers. This framework should rely on a common modeling for services, service enablers, and resources.

The main interest of the proposed approach lies in the identification of the dependencies between the services and the service enablers. This allows better design of the IMS services by defining clearly which service enabler is involved in which service, and how a service enabler is linked to technical functions. This approach optimizes the treatment of service interaction between IMS service enablers by tracing the impact on the user perception of the service. It will also enhance service management aspects by detecting how the failure of one or many technical functions can affect service enablers and the use of the IMS service. It is a tool to identify the user personal information that should be shared between services, to define which service enabler is responsible for which information, and then to design services that reuse this personal information through these service enablers.

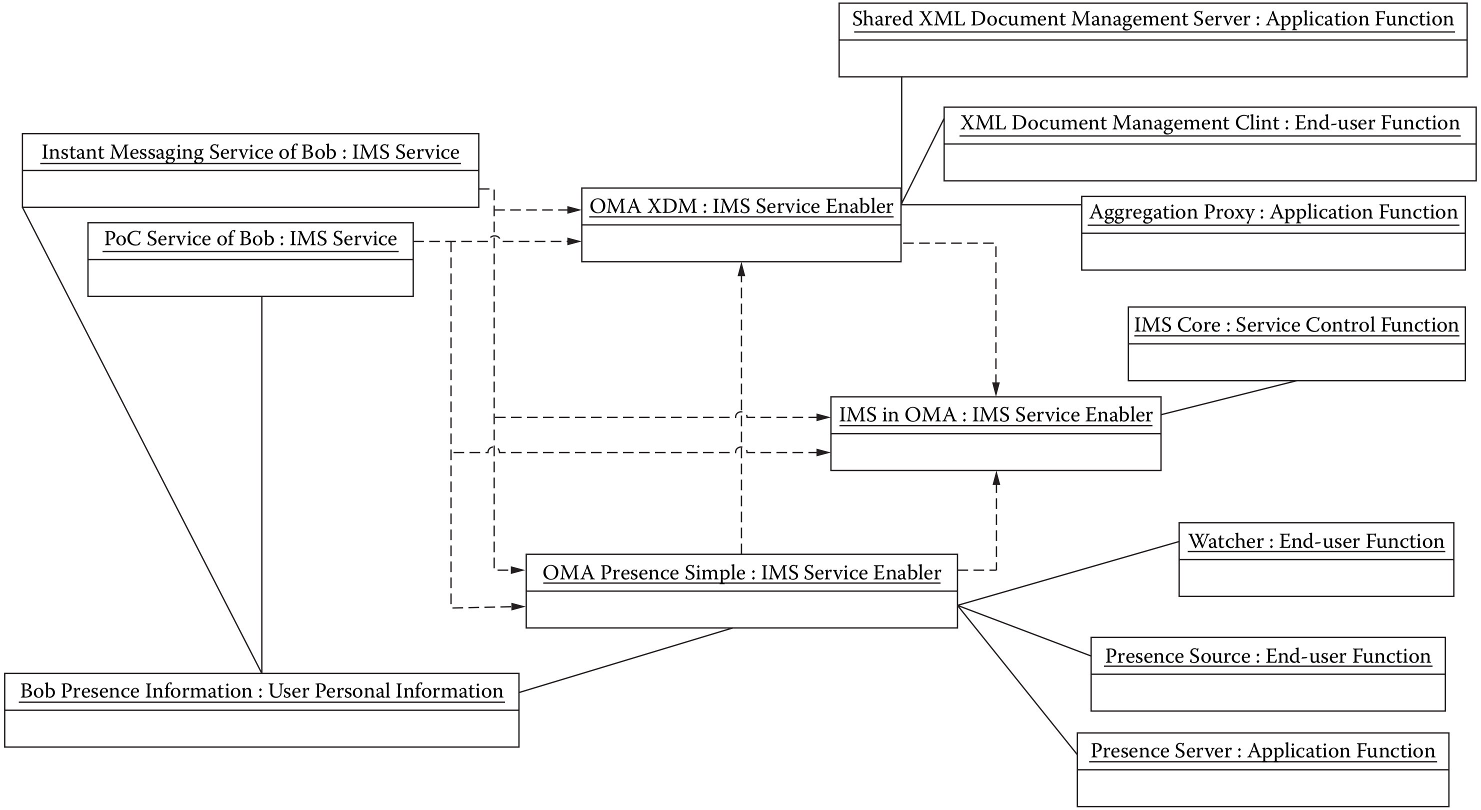
Shared XML Document Management Server : Application Function
XML Document Management Clint : End-user Function
Instant Messaging Service of Bob : IMS Service
OMA XDM : IMS Service Enabler
Aggregation Proxy : Application Function
PoC Service of Bob : IMS Service
IMS Core : Service Control Function
IMS in OMA : IMS Service Enabler
Watcher : End-user Function
OMA Presence Simple : IMS Service Enabler
Presence Source : End-user Function
Bob Presence Information : User Personal Information
Presence Server : Application Function

FIGURE 1.12
Relationship and dependencies of XDM, IMS, and presence simple service enablers.

Glossary

3GPP:	3rd Generation Partnership Project
API	application programming interface
AS	application server
CSCF	call state control functions
DSL	digital subscriber line
GSM	global system for mobile communications
GUI	graphical user interface
HSS	home subscriber server
IETF	Internet Engineering Task Force
iFC	initial filter criteria
IMS	IP multimedia subsystem
IN	intelligent network
INAP	intelligent network application protocol
ISDN	integrated services digital network
ISUP	ISDN user part
IT	information technology
ITU	International Telecommunication Union
NGN	Next-generation networks
OMA	Open Mobile Alliance
OSE	OMA service environment
PSTN	public switched telephone network
S-CSCF	serving call state control functions
SIB	service independent building block
SIP	session initiation protocol
SIM	subscriber identity module
SPT	service point trigger
TISPAN	telecommunication and Internet converged services and protocols for advanced networking
UMTS	universal mobile telecommunications system
WLAN	wireless local area network
XML	extensible markup language
TMF	TeleManagement Forum

References

1. Arbanowski, S. et al. 2004. I-centric communications: Personalization, ambient awareness, and adaptability for future mobile services. *IEEE Communications Magazine* 42(9):63–69.

2. Knightson, K., N. Morita, and T. Towle. 2005. NGN architecture: Generic principles, functional architecture, and implementation. *IEEE Communications Magazine* 43(10):49–56.
3. 3GPP. IP multimedia subsystem (IMS), TS 23.228.
4. Bertin, E., E. Bury, and P. Lesieur. 2003. Operator services deployment with SIP: Wireline feedback and 3GPP perspectives. ICIN 2003, Bordeaux, April 2003.
5. Schulzrinne, H., and J. Rosenberg. 1999. Internet telephony: Architecture and protocols—An IETF perspective. *Computer Networks and ISDN Systems* 31(3):237–255.
6. Tang, B. Y. C. 2005. Evolving to wireless and wireline convergence—An overview of IMS. Wireless and Optical Communications, 2005. 14th Annual WOCC 2005, 27, April 22–23.
7. Marquez, F. G., M. G. Rodriguez, T. R. Valladares, T. de Miguel, and L. A. Galindo. 2005. Interworking of IP multimedia core networks between 3GPP and WLAN. *IEEE Wireless Communications* 12(3):58–65.
8. Lin, F. J. 2005. A survey on wireless/wireline integration. Wireless and Optical Communications, 2005. 14th Annual WOCC 2005, 26, April 22–23.
9. 3GPP. IP multimedia session handling; IM call model, TS 23.218.
10. Schilit, B. N., D. M. Hilbert, and J. Trevor. 2002. Context-aware communication. *IEEE Wireless Communications* 9(5):46–54.
11. Raento, M., A. Oulasvirta, R. Petit, and H. Toivonen, H. 2005. ContextPhone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing* 4(2):51–59.
12. Bertin, E., E. Bury, and P. Lesieur. 2002. Next-generation architectures: Which roles for an incumbent operator? Proceedings of the Eurescom Summit 2002.
13. Bertin, E., E. Bury, and P. Lesieur. 2004. Intelligence distribution in next-generation networks, an architectural framework for multimedia services. IEEE International Conference on Communications, ICC 2004, Paris.
14. Gouya, A., N. Crespi, and E. Bertin. 2006. SCIM (service capability interaction manager). Implementation issues in IMS service architecture. *IEEE International Conference on Communications.*

AU: pls supply location/dates of conf.

15. Carugi, M., B. Hirschman, and A. Narita. 2005. Introduction to the ITU-T NGN focus group release 1: Target environment, services, and capabilities. *IEEE Communications Magazine* 43(10):42–48.
16. OMA. OMA service environment. Approved version 1.0.4, 01 Feb 2007, OMA-AD-Service-Environment-V1_0_4-20070201-A.
17. 3GPP. Presence service using the IP multimedia (IM) core network (CN) subsystem; TS 24.141.
18. 3GPP. Messaging using the IP multimedia (IM) core network (CN) subsystem; TS 24.247.
19. 3GPP. Conferencing using the IP multimedia (IM) core network (CN) subsystem; TS 24.147.
20. OMA. Dictionary for OMA specifications. Approved version 2.6, Jun. 2007, OMA-ORG-Dictionary-V2_6-20070614-A.
21. Ben Yahia, I., E. Bertin, N. Crespi, and J. P. Deschrevel. 2006. Service definition for next-generation networks. International Conference on Networking. ICN'06, Mauritius.
22. Lovelock, C. 2001. *Services marketing, people, technology, strategy,* 4th ed. Englewood Cliffs, NJ: Prentice Hall.

23. Grönroos, C. 2000. *Service management and marketing: A customer relationship management approach,* 2nd ed. Chichester, UK: John Wiley & Sons.
24. 3GPP. 2005. 3GPP definition, TR 21.905, V6.7.0.
25. TMF Forum. Shared information and data (SID) model. GB922 and addenda, release 7, January 2007.
26. Keck, D. O., and P. J. Kuehn. 1998. The feature and service interaction problem in telecommunications systems: A survey. *IEEE Transactions on Software Engineering* 24(10):779–796.
27. OMA. OMA push to talk over cellular (PoC). Approved enabler version 1.0.2, September 2007.
28. OMA. OMA XML document management. Approved enabler version 1.0.1, November 2006.
29. OMA. OMA presence simple. Approved enabler version 1.0.1, November 2006.
30. OMA. IMS in OMA. Approved enabler version 1.0, September 2005.
31. OMA. OMA device management. Approved enabler version 1.2, February 2007.
32. Ryu, S. et al. 2005. Research activities on next-generation mobile communications and services in Korea. *IEEE Communications Magazine* 43(9):122–131.