Which architecture for integrated services?

Emmanuel Bertin, Pascal Lesieur
France Telecom R&D
Caen, France
emmanuel.bertin@francetelecom.com

Abstract

The integration between data services and communication services (relying on Next-Generation Networks) is a need for the users in order to simplify the usage of their multiple services. Several options are possible to build integrated services: integration on the interface layer / presentation tier, integration on the application layer / business tier and integration on the application data layer / data tier. We present here these options and illustrate them with two prototypes achieved in our labs.

1. Introduction

Telecommunications services are facing major technical evolutions with Internet telephony growth and evolutions toward NGN (Next-Generation Networks). Their usage is also evolving toward user-centric [1] and integrated services, relying ubiquitously on multiple access networks and offering convergence between multiple services (e.g. voice/video connectivity, community tools, presence, conferencing, gaming, TV broadcasting...) [2]. Operators, application service providers and even software editors are partially targeting similar markets with similar services (e.g. voice over IP, instant messaging, address book...).

In this context, several architectures may be chosen to provide integrated services to the user: the integration may be realized on several architectural layers and with various mechanisms. The aim of this article is to detail and evaluate the main architectural scenarios to implement the convergence between data and communication services into integrated services. We define an integrated service as a telecom service that both:

- Provide usage continuity between data and communication service features (e.g. between address book and call forwarding);
- And provide the same usage whatever is the access network from the user and from his contacts.

Layered models are used both for data and communication services, but these models are not usually compared and interrelated. We investigate first how the data services and communication services may be represented with a layered model. Then we study the relationships between these models and the options for service integration. Finally, we illustrate two options of service integration with prototypes trialed in our labs.

2. Layered models for data and communication services

2.1 Layered model for data services

The goal of data services is to provide data that are treated to fulfill user needs, according to business logic. Provided data are typically multimedia content like formatted text, images, sounds, video... Data services are following the “dumb network paradigm”, where the core network has no functional role for the service, except the data transport [3]. Concerning the application entities, two major paradigms are used: client/server or peer-to-peer. In both cases, the end-device (e.g. a PC) is directly in communication with applications for the execution of the services and the network entities are only involved in the routing of the data flows (and absolutely not for the service triggering). All the information about the service session is stored in the end-devices and in the applications. We will here only focus on the client/server model, where a client residing in the end-device communicates with a server to provide services to end-users.

When using the client/server paradigm for data services, the "N-tier" model (where "N-tier" means any number of levels serving separated tasks) provides a commonly used architectural model [4]. This model enables developers to create flexible and reusable application by breaking data services into tiers. If they decide to change of technologies or to scale up, developers should only have to modify a specific layer, rather than having to rewrite the entire application. To
achieve this, each layer should exchange information only with the layers above and below it. Each layer should thus have clearly defined interfaces (API) toward upper and lower layers. In addition, each layer should be able to exist on physically different systems to allow an independent implementation of each layer. However, the "N-tier" model is more of a guideline than a specification and has to be adapted to each application constraints (e.g. performance, re-usability, development time…).

A typical "N-tier" model with 3 tiers is described below. Two usual options are the light client and heavy client architectures. These two options are shown in the figure 1. The network cloud indicates an intranet or internet network. It materializes also the separation the clients and the servers.

<table>
<thead>
<tr>
<th>Presentation Tier</th>
<th>Web browser</th>
<th>Heavy client</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Web server</td>
<td></td>
</tr>
<tr>
<td>Business Tier</td>
<td>Business objects and rules</td>
<td></td>
</tr>
<tr>
<td>Data Tier</td>
<td>Data storage</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 1. Layered model for data services**

In this figure, the roles of each tier are the followings:

- The presentation Tier provides an interface for the end user into the application: it transforms the results of the Business Tier into something usable by the end user. It can be achieved with a heavy client, or with a web browser that parses content from the web server to deliver a human friendly user interface.
- The Business Tier contains the service logic and the rules to manipulate the data. It is structured with objects, called business objects, which model the logic of the application (e.g. EJB or COM objects). For example, an online shopping application could contain a "Product" business objects, with "getDescription", "getPrice", "setDescription", "setPrice" methods.
- The Data Tier is the Database system (e.g. SQL Server), that is intended to deal with the storage and retrieval of information (and not with manipulation or delivery of information). It is structured with the data model of the service.

### 2.2 Layered model for NGN communication services

Communication services are moving from legacy telephony to multiple instant communication services: chat, instant messaging, voice, video, presence… A typical service example is the seamless use of multiple communication modes (chat, instant messaging, voice, video), through multiple access networks and terminals, and providing an availability management in coherence with the user's communities [5].

The goal of communication services [6] is to establish a communication session between users or to modify the basic session establishment process. The session manipulated by instant communication services are typically voice sessions, but also video sessions, instant messaging sessions or collaboration sessions. The term session is not understood here with a technical meaning, but from a user perspective, as an interactive exchange between 2 or more persons in order to communicate.

As for data services, the architecture of communication services is layered. We will not detail here the layered architecture of the legacy communication services (e.g. telephony), but we focus only on the new communication services relying on Next-Generation Networks (NGN). ITU-T has defined the principles for the NGN [7]. These principles have been implemented by 3GPP with IMS (IP Multimedia Subsystem) [8], using the SIP protocol. IMS as defined by 3GPP is suitable for mobile networks. Within ETSI TISPAN project, IMS extensions have been standardized for other types of access networks (xDSL). The IMS rely on a formal separation (through standard protocols or APIs) between:

- The transfer layer: various access networks (UTRAN, WLAN, xDSL) connected to a single backbone
- The control layer: control functions designed to be common to these various access networks (e.g. network attachment control, resource and admission control, session establishment control, service triggering control).
- The application layer: application functions for access-independent session-based services (i.e. that are triggered during a session) that contain the service logic and the manipulation rules in order to modify the basic session establishment process (e.g. with transfers, callback, reachability, call log…).
Concerning the IMS, major elements related to services [9] are:

- **S-CSCF (Serving Call State Control Functions):** the session control and service triggering entity
- **HSS (Home Subscriber Server):** the central service and network database
- **AS (Application Server):** service platforms providing session-related services to users (presence, push-to-talk, conferencing…). AS offer APIs like OSA/Parlay or SIP servlet for application execution.

As we are not dealing here with networking issues but only with service integration issues, we do not need to consider the transfer layer. In order to allow the comparison with data services, we propose here to add two others layers to the control layer and the application layer: an interface layer and an application data layer.

The end-device includes not only the transport functions to connect to a network, but also the service interface part. We propose to define the interface layer as the interface provided to the end-user in order to initiate and participate to a session: it performs the transformation of the technical messages coming from the application layer (through the control layer) into something understandable and usable by the user (and vice versa). For example, an interface for the presence service will transform the presence protocol messages into a user interface displaying the presence of user's buddies. This layer is necessary because, like for data services, the user interface is gaining more and more importance for instant communication services.

We propose to define the application data layer as the layer that is responsible for the storage and retrieval of information manipulated by the applications, like for example presence and availability information, contact list information, call forwarding information, user preferences information… It is structured with the data model of the services. The formalization of this application data layer enables to identify precisely the issues for the integration of data from different communication services.

### 3. Options for service integration

The layered models for data services and for communication services can be compared layer by layer. The interface layer is similar to the presentation tier. The application layer is similar to the business tier. The application data tier is similar to the data tier. Only the control layer does not have any equivalent in the data services. This is due to the lack of service control and service triggering mechanisms in the Internet paradigm, as seen in §2.1. As the data services do not provide the equivalent of a control layer, integration between data and communication services can not be achieved using this control layer.
Integration of data and communication services may be achieved on each architectural tier. The mechanisms that can be used for service integration are different on each tier:

- **Service integration on the Data Tier / Application data layer** means that a common database scheme is shared between all the services to integrate. This provides a very tightly-coupled integration, where the representation of all data is truly common, but with a major lack of flexibility because the database scheme should evolve to integrate any new service.

- **Service integration on the Business Tier / Application layer** implies to share business objects and data manipulation rules between the services to integrate. This could be achieved in a loosely-coupled way by using the SOA (Service Oriented Architecture) paradigm [10]: each service is designed as a set of elementary self-contained services (e.g. web services) that expose in and out interfaces and that have a clear responsibility on the data manipulated. The integration between services could then be achieved by the composition of these elementary self-contained services into integrated services [11]. In the field of communication services, the decomposition of services into service enablers aims at the same goal. This approach enables quicker service integration, but implies still a substantial development process to integrate a new service. It implies also a shared knowledge from the semantic of the interfaces exposed by the elementary services and the setup of a middleware communication bus between these services.

- **Service integration on the Presentation Tier / Interface layer** signifies the aggregation of the interfaces from several services into the same client, without modification of the business logic / service logic of each service. It can be realized differently for light or heavy clients. For heavy clients, plug-ins should be developed for each service. These plug-ins may either replace the heavy client and contact directly the business tier, or may be built upon APIs from the heavy client. All these plug-ins can then be integrated in a common GUI to provide integration and usage continuity between the services. This approach does not imply developments on the business tier, but it implies the development and deployment of plug-ins on all the end devices. Concerning the light clients, the service integration may be achieved by aggregating several service interfaces in the same GUI using web mechanisms like portlet or widget. This approach simplifies the development process to integrate services and enable flexible integration of new services, but the level of integration between the functionalities of each service remains low. However, as the communication clients follow today the heavy client architecture (because the messages exchanged with the control layer are bidirectional), the service aggregation through light clients can not be fully achieved today.

### 4. Experimentations of service integration architectures

In order to compare these approaches with operational results, we describe here two prototypes experimented in our labs: the NCOM (i.e. eNabler of COMmunications) middleware and the Adcom (i.e. advanced communications) platform.

#### 4.1 Integration on the presentation tier / interface layer

NCOM middleware focus on the integration of several communication services on the presentation tier / interface layer. The aim is to offer the possibility of a convergent usage between services and not only the juxtaposition of these services. Our approach was to implement an integration middleware on the workstations. NCOM middleware enable the integration of services that are already installed on the workstation and that remain usable without it. The NCOM middleware is installed only on the workstation and it installation does not imply any modification on the service platforms. NCOM facilitates communicative usages by automating several actions like for instance click to dial or click to conference and by allowing switching from a communication service to another, like for instance switching from a chat session towards a voice call.

The NCOM middleware is composed of plug-ins for various communication or data services:
- An IP telephony plug-in which uses the API of a SIP softphone connected to an IPBX
- An IP telephony plug-in which uses the API of a MGCP softphone connected to an IP Centrex
- A presence plug-in connected to a corporate LCS Microsoft server, using SIMPLE protocol
- A presence plug-in connected to a corporate Jabber server, using XMPP protocol
- A conferencing plug-in connected to a web conferencing system using Microsoft Netmeeting
A conferencing plug-in connected to an operator conferencing service
Several click to dial plug-in using several PBX APIs
A directory plug-in connected to an LDAP corporate directory

All these plug-ins where integrated in a dedicated User Interface which presents the user availability on different media (in fact the availability on the different plug-ins) and enable to initiate multiple communication actions (conferencing, click to dial…)

Business Objects
Data Storage
Service Logic
Service Data
Session Establishment

Figure 5. Integration on the presentation tier / interface layer

The advantage of this approach is that each communication or data service relying on a heavy client on the workstation can be completely or partially integrated. This is done by implementing a plug-in for this service that will comport toward the service platform as the heavy client. On the other side, the integration will be limited to the functionalities implemented in the NCOM middleware.

The main drawback in this approach is the installation and maintenance on the workstation. The middleware must be conformant with a specific version of an operating system and must evolve with each roadmap of the services it ingrates. This middleware is thus very dependant of external factors: the change of the client interface of one of the services, the adding of new functionalities (either in the OS, in the office tools or in the services) may impact the NCOM middleware (e.g. because of the lack of some new functionality or because of interface incompatibility).

4.2 Integration on the data tier / application data layer

Adcom is a Multimedia IP Centrex platform developed for the large enterprises of the corporate market. The supported services are voice and video communications, instant messaging, multimedia conferencing, call forwarding, unified reachability, address book and contact list, unified call log.

All these services are integrated at the data tier / application data layer and at the business tier / application layer. This means that all the services are using the same unified data model. This data model is based on the identity of the users and on the enterprise numbering plan (each enterprise owns its private numbering plan which can overlap another numbering plan belonging to another enterprise). As the data model is common between all services, the provisioning is done in a unified way. All user data (e.g. user identity, availability state, used terminals, preferences…) are thus shared between all services.

The user benefits from the integration of the service logics with convergent services like multimedia conference (voice, video, web) or unified presence and availability management. The user’s availability can be automatically modified if a user is in a call or in case of inactivity (duration setup by the user). His buddy list informs him about presence/activity information of his contacts in real time. Even if he uses several endpoints, he can give to his contacts one unique phone number for all his devices (PC, IP Phone, mobile phone). He keeps his phone number and all his services whatever the access network (LAN, DSL, Wifi, GPRS, and UMTS) by working with different office tools:

- PC + IP Phone on the desk
- Laptop + Soft Phone out of the office
- Smartphone connected via GPRS network

All the communication media are managed by a software client available over Windows platforms as on smart phones and which allows defining rich call routing:

- Simultaneous / Sequential call towards fix and/or mobile phone
- Up to the caller number / presence state / date

The Adcom platform provides thus tight integration between communication and data services, with a common data model and common service logic.
This approach leads to a well integrated multimedia convergent solution. But the adding a new service would mean changes in the data model and modifications in the service logic or in the business objects. Thus it is not realistic to envision the integration of a new data service in such an environment. Only service that were initially taken into account in the service platform can be integrated.

5. Conclusion

The options we have presented have each advantages and drawbacks when facing to operational problematic. Deploying service integration solution has some major stakes, including at least:

- Taking into account the existing services (PBX, directory, IM system, other services)
- Ability to propose unbundled offers with the services and not only one integrated service
- Ergonomic integration in the office software
- Installation and maintenance on the end-devices
- Enabling the use of a wide range of devices (wireline phones, wireless phone, desktop PC…) with a unified reachability management

The choice of the approach to integrate data and communication services is thus related to the objective of the integration.

Integration on the presentation tier / interface layer is very optimal:

- to interoperate with different existing services, because a new service can be easily integrated without modifying the business objects and data logic
- to bundle / unbundled service, because several integration bundles can be easily built
- to integrate office software because the presentation layer can be plug into office tools with plug-ins

But other approaches are more efficient

- to simplify the installation and maintenance for the client software because the client software to install is always the same
- to use multiple devices with a common reachability management, because common service logic and business rules may be applied for these various devices. Indeed, the integration on interface layer / presentation tier does not enable to use fully routing and service triggering mechanisms implemented in the control layer (for instance in case of fix/mobile convergence).

In conclusion, the integration on the presentation tier / interface layer is probably the best suited solution where flexibility and evolutivity is needed.

References

[9] 3GPP, "IP Multimedia Subsystem (IMS)", TS 23.228, Release 6

0-7695-2622-5/06/$20.00 (c) 2006 IEEE