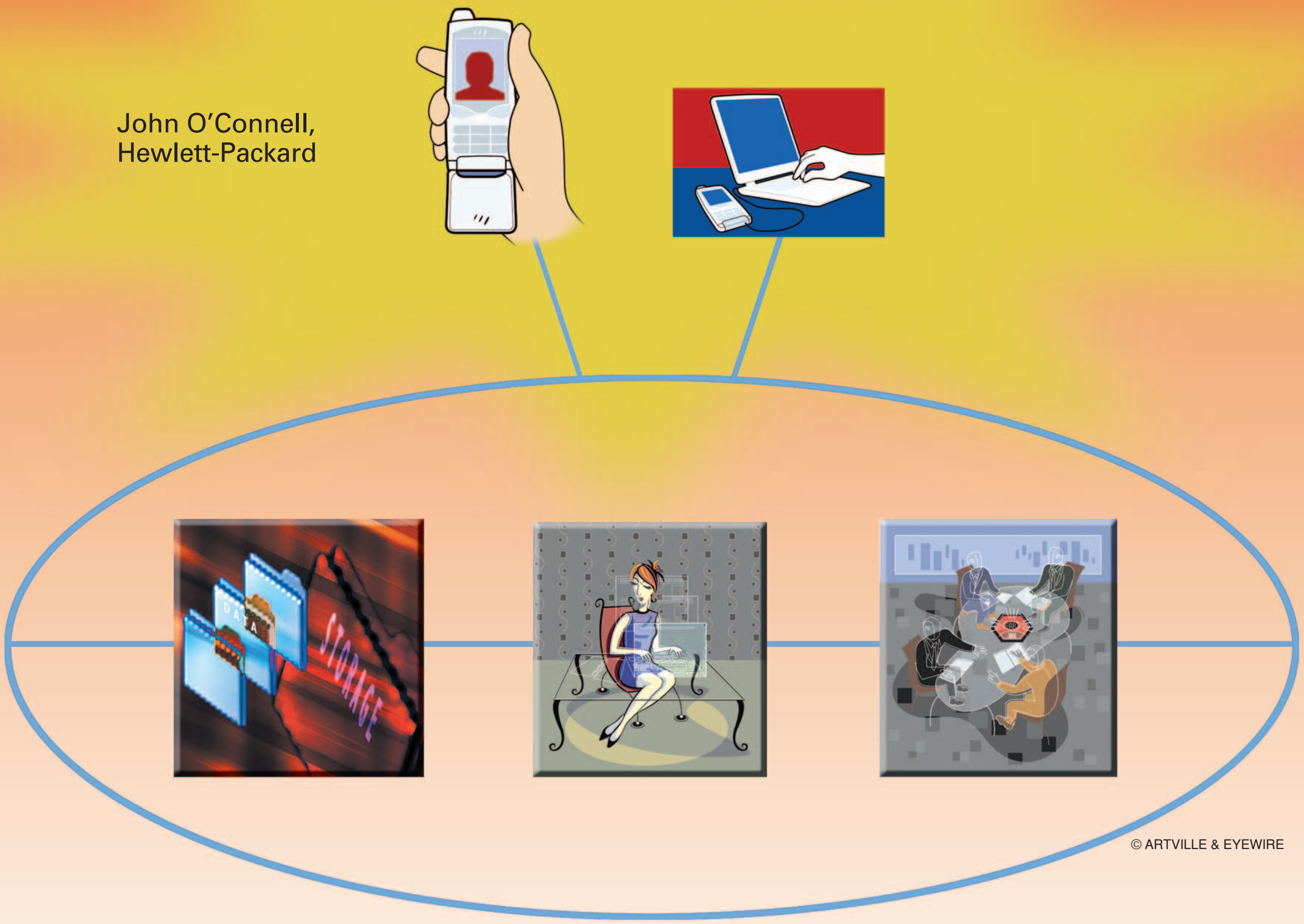
Service Delivery within an IMS Environment

John O'Connell,
Hewlett-Packard



© ARTVILLE & EYEWIRE

Abstract: *The IP Multimedia Subsystem (IMS) standard is being widely adopted by both wireline and wireless service providers to support the next wave of multimedia communication applications. As currently defined, that IMS standard proposes a detailed architecture for the transport and control layers of the next-generation telecommunication networks, but leaves many questions unanswered on how the application layer should be structured. A more structured approach to service delivery will be required in order to deliver new multi-media applications in an efficient and cost-effective way. An approach based on Service Delivery Platform (SDP) proposes a layered service architecture, with a strong emphasis on re-use of common service enablers and on open APIs. This paper describes work being undertaken within HP's IMS Experience Center to validate a horizontal service delivery architecture for next-generation IP multi-media applications.*

Drivers for IP Multimedia Applications

Increased competition and price pressure is directly impacting the revenue streams of traditional telecom service providers. The roll-out of wireline and wireless broadband access networks has lowered the entry barrier for VoIP service providers, allowing them to offer lower-priced telephony services to consumer and enterprise customers. As the average revenue per user for digital telephony declines (on both fixed and mobile networks), many service providers are looking to the new wave of IP-based multi-media communication applications to replace that lost revenue. This has caused a strong shift in focus within the communication industry from digital telephony to more media rich applications.

This shift is being driven by many factors. Mobile phones, portable PCs and other personal media devices all now include multimedia capabilities and support media capture and playback features as standard. Wireline and

Digital Object Identifier 10.1109/MVT.2007.898099

1556-6072/07/$25.00©2007IEEE

wireless broadband access networks have become ubiquitous, whether based on DSL, WiFi or 2.5G/3G. Also, as media becomes digital, it becomes much easier and more natural to inject content, whether private content or public licensed content, into communication sessions.

User expectations are also changing. Our experiences with the internet are fuelling demand for new ways to communicate, and to share our thoughts and feelings (in the form of photos, video clips, voice messages, etc) in real-time with friends and family. The success of services such as SMS (on mobile phones) and Instant Messaging (on the internet)—and in particular, the instantaneous nature of these services has redefined user's expectations of being able to instantly share "stuff", in real-time, across multiple devices, via any network, and perhaps most importantly, at a reasonable price. Finally, users also expect services to work seamlessly across multiple devices. Content that is generated on a mobile phone can be downloaded and edited on a PC, before being sent electronically to family members who may choose to play it back on a TV or personal video device.

All of these trends are expected to drive demand for a wide range of new applications, such as:

- Content sharing applications, where various types of media objects can be shared in real-time across multiple devices
- Personalised interactive TV services, extending the users' TV experience with value-added services
- Rich call services, which enhance the traditional telephone call with multimedia features
- Instant group communication services which enable multiple forms of interaction within a group or community context.

In anticipation of this shift, many service providers are deploying SIP-based infrastructure, in compliance with the 3GPP's IP Multimedia Subsystem (IMS) [1] standard. IMS brings many potential benefits for service providers, as it defines an access network agnostic architecture for delivering real-time multimedia services (including voice services) over an IP-based network, with built in support for inter-networking, roaming, access control and online/offline charging. IMS is also expected to leverage the latest IT and IP technology, and this is expected to lead to lower cost of ownership, compared to traditional telecom equipment which has often been based on telecom-specific technology.

Despite these many promises and despite the hype surrounding IMS today, the technology will ultimately be judged on its ability to deliver new real-time multimedia applications in a cost effective way. This cost aspect is important because alternative architectures, for example based on peer-to-peer approaches, are also being proposed and adopted by players within the industry. This raises the question of how to structure the IMS service layer so that IP multimedia applications can be created rapidly and can be delivered efficiently and economically.

A Horizontal Service Delivery Architecture

Delivering a wide range of multi-media applications in a cost-effective way will require a more structured approach to service delivery, where service resources and subscriber data are shared across applications. Given the additional complexity that these multimedia applications will introduce, the service architecture that has been adopted for today's 2G/TDM applications is not suitable for next-generation services. The trade-off that was made in 2G/TDM networks, in favour of performance and reliability at the expense of openness and ease of operation, is no longer appropriate. Increased competition and the need to offer innovative services will require service providers to place greater emphasis on open networks (e.g., in support of 3rd-party services) and on flexibility (e.g., in response to user demand for greater control over service offerings). This has led many within the industry to propose a horizontal service delivery architecture for IMS, with a service enablement layer sitting between the control layer and the applications.

Such a layered approach is central to the concepts and principles of a Service Delivery Platform (SDP). As shown in Figure 1, such an architecture addresses the twin goals of sharing common functions and resources across

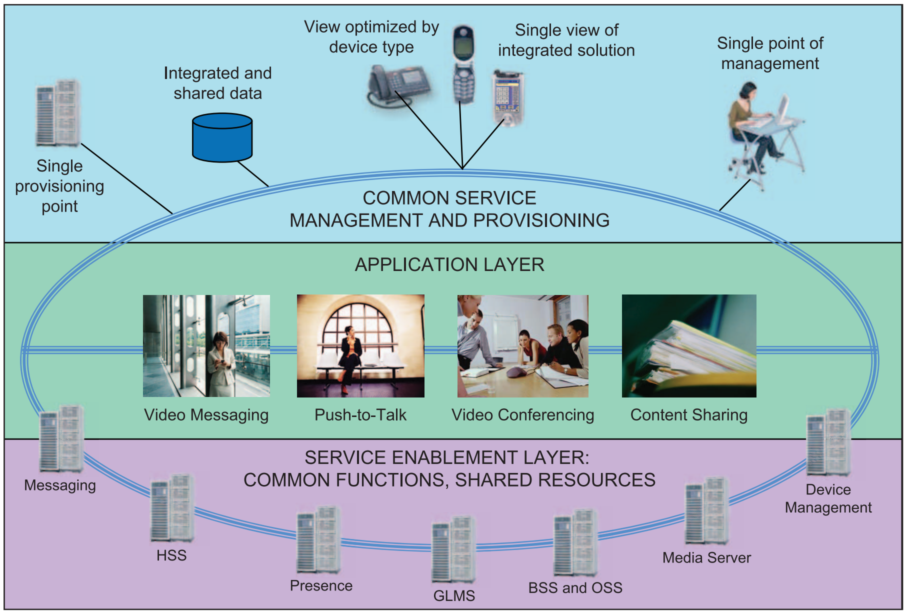

View optimized by device type
Single view of integrated solution
Single point of management
Integrated and shared data
Single provisioning point
COMMON SERVICE MANAGEMENT AND PROVISIONING
APPLICATION LAYER
Video Messaging
Push-to-Talk
Video Conferencing
Content Sharing
SERVICE ENABLEMENT LAYER: COMMON FUNCTIONS, SHARED RESOURCES
Messaging
Device Management
HSS
Presence
GLMS
BSS and OSS
Media Server

FIGURE 1 Horizontal Service Delivery Architecture for IMS.

IMS WAS ORIGINALLY DEFINED BY 3GPP TO SUPPORT REAL-TIME MULTIMEDIA APPLICATIONS WITHIN 3G MOBILE. THE BROADBAND AND CABLE INDUSTRIES HAVE ALSO NOW ADOPTED THE SAME CONCEPTS TO SUPPORT THE DELIVERY OF REAL-TIME MULTIMEDIA APPLICATIONS IN BROADBAND AND CABLE NETWORKS.

multiple applications and of adopting common frameworks for provisioning, management, user access, etc, to avoid unnecessary duplication of functionality. By adopting common functions for service delivery and common frameworks for service management and provisioning, the SDP approach aims to dramatically reduce the incremental cost of introducing a new service in to the network.

Within this layered horizontal architecture, the role of the service enablement layer is to host a set of application-independent building blocks that offer generic functionality to support a diverse range of multimedia applications. Examples of IMS service enablers include presence server, group contact server, media resource function, messaging gateways, user profile server, etc. The ideal service enabler is both multi-network and multi-device, acting as a point of convergence and consolidation in a heterogeneous environment, while offering network-agnostic and device-agnostic APIs to application developers. These service enablers can be deployed once, and then re-used across multiple applications.

A complete service delivery environment also needs to address the operational, provisioning and management aspects of services. In order to avoid unnecessary redundancy and to reduce OPEX, this implies offering common frameworks for provisioning and management, so that a common look-and-feel is provided across a range of services.

Finally, this need for a common look-and-feel must also apply to the end user's view of the services. Many of the new multimedia applications are expected to be complex in nature, but must be easy-to-use in order to offer an attractive user experience. An intuitive and integrated user interface can hide the underlying complexity of the services, and can help to present new services and new service features through a familiar interface. This applies both to the interface that is used to access the service from the user's end device and to any interface that is offered to the user to provision, configure and personalise that service.

Combining IMS and SDP

Both IMS and SDP are being positioned within the industry as key technologies in the migration to NGN, with both claiming to address convergence and service innovation. By convergence, we mean delivering the same service over multiple access networks, with the goal that the service can be deployed once, and made accessible from many devices. By service innovation, we mean enabling the rapid creation and deployment of new applications that deliver new user experiences and that typically are expected to deliver new multimedia and multimodal features. However, despite claiming to deliver the same benefits, it is important to see IMS and SDP as being complementary, where the combination of IMS and SDP can bring many benefits.

On one hand, IMS offers a framework for IP communication applications, across multiple networks and devices, with a well-defined standard control layer which ensures interoperability and roaming (important for mobile and nomadic users), end-to-end session management and a coherent framework for service charging. IMS also promises a clean separation of the application layer from the core network.

In parallel, SDP can bring a lot of value at the service layer, by offering a structured framework for managing the delivery of services, and by doing so in a cost-effective way with a strong emphasis on re-use of common functions across multiple applications. Many of the SDPs in the market also address the issue of how to open up networks to 3rd-party applications, and this more flexible approach can be an enabler for service innovation, as many more players, such as ASPs and 3rd-party content providers, can create and deploy multimedia applications. The use of standard IT development tools, based on familiar java or XML programming methodologies, is also an important aspect of SDP, and can help to reduce the complexity of creating new applications.

Experimenting with IMS Service Delivery

In order to understand the challenges involved in creating and delivering IMS applications and in order to validate this layered approach to IMS service delivery, we have taken a practical approach by implementing a set of IP multimedia applications within a functionally complete IMS environment. HP's OpenCall Experience Center [3], located in Grenoble, has been established to validate this horizontal approach to service delivery for IMS. The center includes a functionality-rich IMS network environment, integrating both HP and partner technology. The center currently hosts an initial set of IMS-compliant service enablers, end devices, and IMS applications, as shown in Figure 2.

This environment is primarily being used to explore the technical issues and implementation choices involved in developing and deploying end-to-end IMS solutions, including both client-side and server-side elements. Elements of the Experience Center include HSS, CSCF simulator, MRF, presence server, XDMS/GLMS, various AS and SDP.

Focus on Instant Group Communication

For this architecture validation work, an example application was chosen. The chosen application is a group-based media sharing application that combines group contact and presence information, media sharing and real-time voice communication within a single session. This application belongs to the category of "instant group communication" applications, which are primarily focused on delivering instant (i.e., real-time) communication-oriented (i.e., person-to-person) behaviour within a group or online community.

These applications aim to deliver new multimedia experiences within a group context, where a group could be a family, a group of friends, social group, work group, or some other form of ad-hoc group that is linked by a common interest. Instant group communication applications build on the success of instant messaging and buddy lists which are now widely used in the internet, and they extend these concepts with new service features to also offer a more media rich and multi-modal experience.

Two important characteristics of this category of applications are the ability to spawn multiple communication services from a contact list or other shared content, and the ability to combine multiple interaction modes, such as voice, video, messaging, content sharing, etc within a single user session.

Given the nature of the applications, it is also very important to present the different service features from within an integrated user interface. A user of the service should be able to browse shared contact information and shared content, to trigger new interaction modes and to inject content into established communication sessions, within a single user session and all through a consistent user interface that offers a common look-and-feel across the different aspects of the application.

In implementing the chosen application within the HP OpenCall Experience Center, two different service architecture models have been adopted and validated. The first implementation approach deploys the application as a standalone element within the IMS network, in compliance with the Application Server (AS) function, as defined by the IMS standard. In this approach, the primary interface between the application logic and the other elements of the solution is a SIP-based interface, although other IP-based and HTTP-based protocols are also used for some parts of the solution. In contrast, the second implementation approach has adopted a web services approach, where the application logic is modeled as a "service chain", invoking a sequence of network functions and network services via XML-based web service interfaces.

By choosing two different implementation approaches, this work not only validates both approaches, but also helps to identify the benefits and challenges that are specific to each approach.

Both implementation approaches, and the lessons learned, are described in the following sections.

IMS Application Server Model

In the first target service implementation architecture, the media sharing application has been implemented as a standalone element within the IMS environment. In this approach, the media sharing application acts as a centralised session controller, with responsibility for co-ordinating session establishment, media mixing and multicasting, addition and removal of interaction modes (e.g., adding/removing voice channels), and session termination. It is deployed as an IMS Application Server, and it interacts with the other elements of the IMS environment using standard protocol interfaces such as SIP, Diameter and HTTP, in order to fulfill these responsibilities.

The core logic of the media sharing application executes within a java execution environment. It interacts

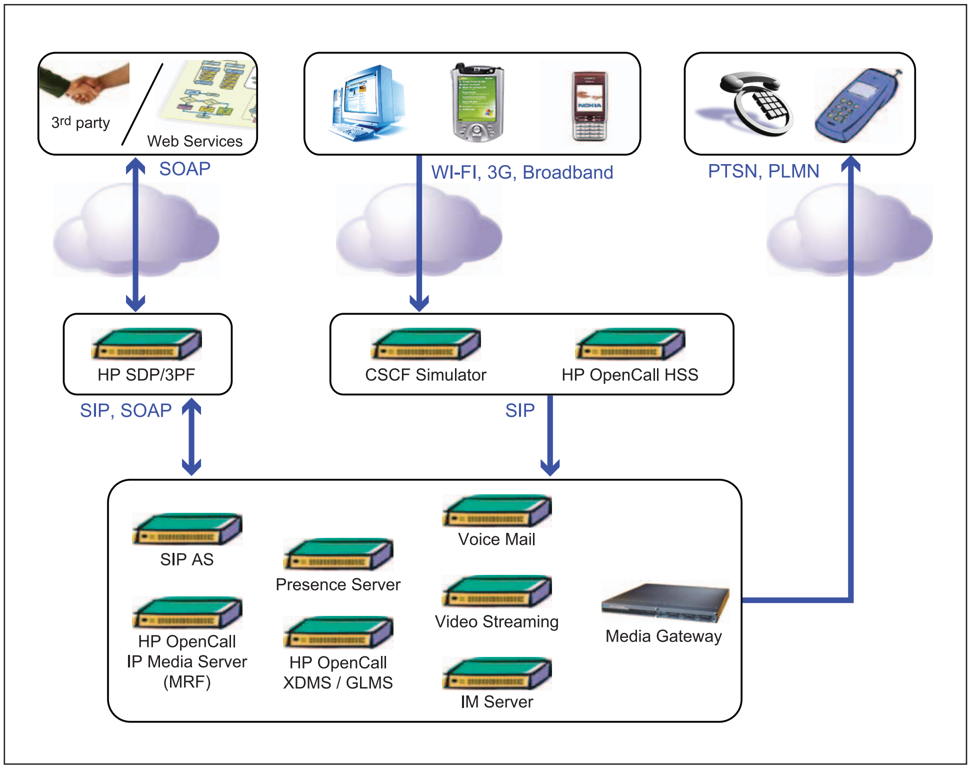


FIGURE 2 Components of the HP OpenCall Experience Center.

TWO DIFFERENT SERVICE ARCHITECTURE MODELS HAVE BEEN ADOPTED AND VALIDATED: ONE DEPLOYING THE APPLICATION AS A STAND-ALONE ELEMENT, AND THE OTHER TAKING A WEB-SERVICES APPROACH.

with the other components of the Experience Center using the protocol interfaces shown in Figure 3.

In a typical use case scenario, the initiator of a session will initially interact directly with the application server in order to configure and establish the group session e.g., choosing who should participate in the session, and with what rights. The Media Sharing application uses both GLMS/XDMS [4] and presence servers to present the session initiator with up-to-date information on which group members are available to participate.

In a messaging and content sharing use case, the session initiator selects the participants, and sends a SIP message with the list of invited participants to the application server. This implementation uses SIP to establish an MSRP-based messaging session, where all messages are forwarded by the application server to the session participants. The media server is invoked, when required, to stream audio or video content to the participants' devices (e.g., when the audio or video content is too large to be downloaded to the end device).

In an instant audio/video conferencing use case, the session initiator again selects the session participants, and in this case, sends an XCAP CPCP request to the application server to launch an instant conference. The application server interacts with the media server to prompt each invitee to join the conference, and once the session is established, the media server is responsible for mixing the voice and video streams and for delivering any shared media content in the correct format to all participants.

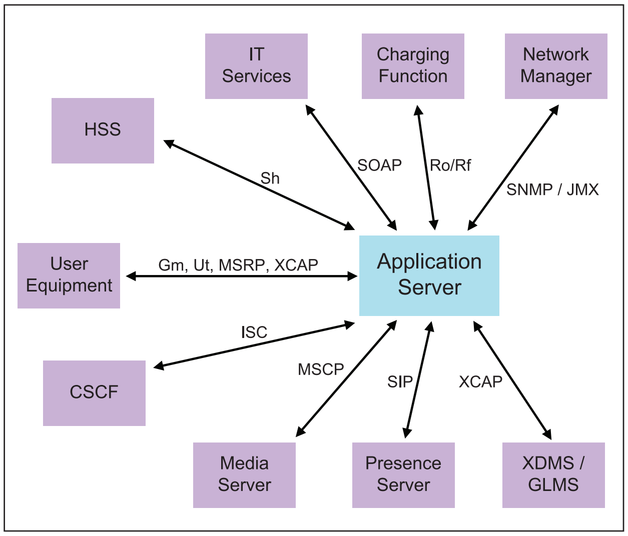


FIGURE 3 IMS-compliant Application Architecture.

Other interfaces that have been included in this implementation include:

- XML/http interface between the Media Sharing server and the Media Server for real-time control of the media streams.
- Sh interface with the HSS, for storing and retrieving user profile and user preference information related to this service.
- Ro/Rf interface with charging engine, for online/offline charging.
- MSRP [2] interface, implemented directly by the Media Sharing application, for messaging-based content sharing between session participants.
- Service management interfaces, based on SNMP and JMX.

Results and Lessons Learned

This work successfully demonstrated the implementation of an end-to-end instant group communication application in compliance with the 3GPP IMS specifications. The chosen implementation architecture conforms with the objectives of a horizontal service delivery architecture (as outlined earlier) in that the implementation relies on application-independent service enablers (GLMS/XDMS, presence server, media server, client software, charging function, etc.) to fulfill many of the basic features of the application. In effect, the application itself acts as a "session orchestrator", invoking these basic service enablers as appropriate (in response to user requests, network-generated requests, and other external events), to deliver the required end-user experience.

It is interesting to note that although the architecture has been presented here as being application server centric, client-based software also played an important role in the implementation. Indeed, much of the development and testing effort was focused on the device side. Application logic was required on the client to deliver the required integrated user interface, and was also required to manage local content. A group-aware application integration framework was implemented for the client devices as a generic plug'n play layer for invoking various network applications from within a single user session. While the media sharing application described here was implemented as a single application, it is expected that multiple such applications, offering different sets of capabilities, will be deployed in the emerging IMS networks. As each new application is deployed within the network environment, a corresponding client application will need to be deployed on the user devices. A generic plug'n play framework on the client can help to reduce the complexity associated with the deployment of client-based application logic.

This project highlighted a number of areas where there is a lack of industry-wide standards today, and where, as a result, pre-standard implementations were developed in this project. Some areas are highlighted here.

A media server control and media resource management framework is required, to complement and extend the current specifications of the Media Resource Function (MRF). The evolution from a voice-centric to video-centric world will change how media resources are used within the network, and this needs to be taken into consideration in the ongoing AS—MRF standardisation work. An XML approach, based on VoiceXML (with video extensions) and CCXML technology, was used in this implementation, to allow direct control of the media server by the centralised application logic.

The project also highlighted the need for a network-wide subscriber data management infrastructure, where both static user data (e.g., profile and preference information) and dynamic user data (e.g., presence and location information) can be accessed easily and efficiently. Such a framework should expose a common user profile data model and an abstract data access API both to provisioning applications and to network applications, ensuring a clear separation of application logic from the underlying data management architecture.

A similar challenge exists in terms of charging APIs, because of the multiple billing systems and billing models that need to co-exist. This implementation adopted a Parlay-like charging API, with support for group billing models, as an abstract API between the application logic and the protocol charging interfaces.

Service Chaining Model

As an alternative to the service implementation architecture described above, we have also developed this instant group communication scenario using a Service Chaining architecture based on web services technology. In this second approach, the set of network functions are exposed as web services and the application logic is implemented as a sequence of web service invocations.

The concept of a Service Delivery Platform—a horizontal service delivery architecture based on web services technology and Service Oriented Architecture (SOA) concepts—is well understood within the industry today. In compliance with the OMA Open Service Environment (OSE) [5], HP's Service Delivery Platform [6] exposes network level functions as web services, and provides policy management and policy enforcement functions to control access by 3rd-party applications to these network functions. Service Chaining extends the SDP concept by providing an environment to facilitate the rapid development of value-add applications that are structured as a sequence of web service invocations. The Service Chaining environment includes service creation tools to specify and validate the application logic (i.e., to define and test the sequence of web service invocations, prior to deployment), as well as a service chaining server that manages the execution of the application logic; that is, it manages the execution of a "service chain".

IN THE SIMPLEST MODEL, SERVICE CHAINING INVOKES SERVICE INSTANCES IN SEQUENTIAL ORDER, TRANSFERRING SESSION CONTROL AND SESSION STATE INFORMATION FROM THE CURRENT SERVICE INSTANCE TO THE NEXT ONE IN THE CHAIN.

A complete service chaining architecture must address multiple requirements:

- It must provide a framework to combine multiple individual network services to deliver a richer, multimodal user experience
- It must enable the seamless transition from one network service to another, maintaining both user sessions and session information
- It must work across multiple devices and multiple networks
- It must provide an integrated user interface, supporting sequential & parallel invocation of services within a single user session
- It should use web technology to allow easy customisation of the service chaining logic

In the context of instant group communication services, a service chain should enable, for example, a real-time discussion between friends to be transferred seamlessly from an instant messaging session to a voice session without having to tear down and re-establish the communication sessions. Similarly, as shown in Figure 4, a service chain can invoke contact list and presence/location information initially, before invoking an instant messaging service, an audio/video conferencing service, or both.

In this service chaining model, the decision to switch from one service to another (that is, to transition to the next service in the sequence) can be triggered by user

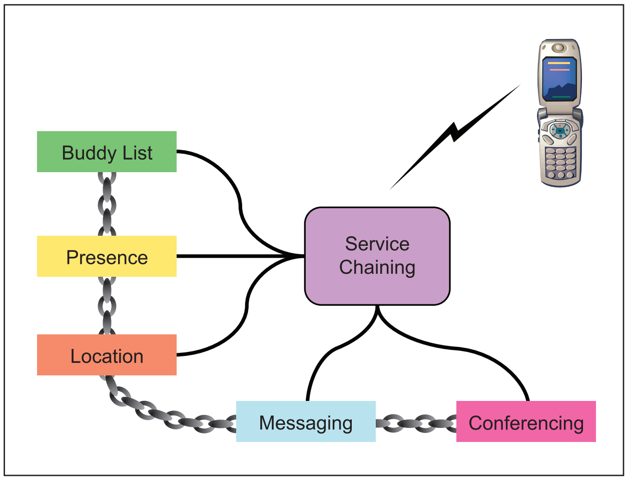


FIGURE 4 Service Chaining Scenario.

requests, by network events, or by other external events. Analogous to the first implementation architecture described above, in this model the service chain application logic acts as the session orchestrator, invoking the appropriate web service in response to external requests.

Service Chaining Architecture

As shown in Figure 5, the implementation of the service chaining architecture relies on three key technology components of hosted within the Service Delivery Platform:

1. The Service Registry which holds information on the underlying network functions that are available, and that can be invoked via web service interfaces
2. The Context Repository, which holds context information for a given user or group session, so that this context information can be maintained for the duration of the service chain.
3. The Service Controller which executes the service chaining application logic that invokes network functions (via their web services interface) in response to user requests and other events.

As each web service in the chain corresponds to the invocation of an independent network function, one of the main challenges for the service chaining architecture is how to ensure a seamless transition from one web service to the next. To achieve this, network session information, session state information, user information, and other contextual information is maintained in the Context Repository independently of each given web service invocation. For example, session-specific contact information that is extracted from the buddy list function (GLMS) can be stored in the Context Repository and then delivered as input to other network functions as they are invoked by the Service Controller in response to user requests. That is, once the user has chosen the list of friends with whom he/she wants to communicate, that subset is stored in the Context Repository. When a user requests the messaging or conferencing functions, the service chaining application retrieves that context information (in this case the buddy list) and passes the information on to the requested services. Other session state information can also be shared in this way between the elements of the service chain.

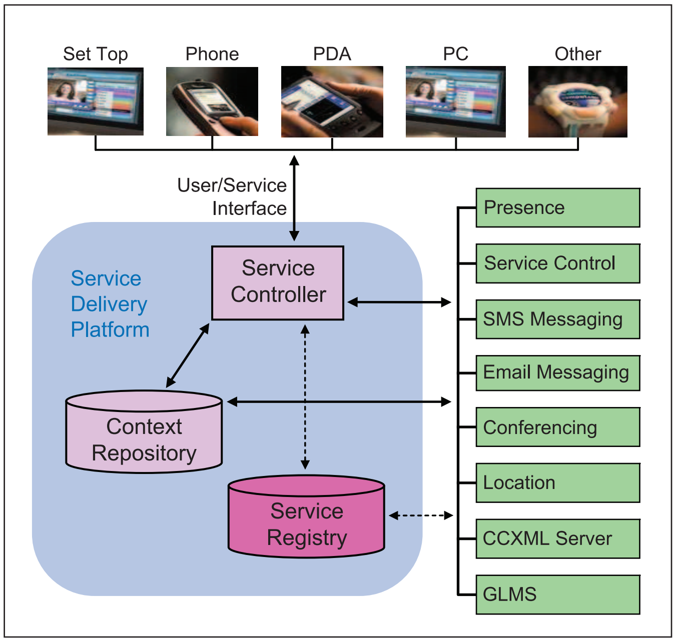


FIGURE 5 Service Chaining Architecture.

Results and Lessons Learned

Note that the use of a Service Chaining implementation model is not restricted to IMS networks or to instant group communication applications. Network functions that reside in existing TDM, 2G and VoIP networks can also be exposed as web services within an SDP environment, and as a result, can be invoked from within service chaining applications. In this way, the service chaining architecture can act as an enabler for the convergence of TDM and IP networks, by facilitating the development of end-user services that span multiple networks and that bundle network services from different sub-networks. For service providers, this allows them to re-use their so-called legacy network functions to deliver value-added applications, even when those network functions have been implemented as vertical silos or have only been deployed within legacy networks.

The core application logic, as specified by the service chain code, plays an important role in managing the user interface of the service. That is, all user interactions are processed by the service chaining logic, which then takes the appropriate action, such as invoking a network function via its web service interface, or transitioning to the next web service in the service chain. In this way, the service chaining application can take responsibility for the user experience, and can offer the same user experience across different device types. As the range of handsets grows more diverse, and with no global standard in place to ease the burden that service providers face each time they seek to deploy a new service, this service chaining approach can remove some of the complexity by offering a clear separation of the user interface aspects from the network functions that are used to deliver end-user services.

This service chaining approach offers great flexibility, allowing easy customization of the application logic on a per-user, per business customer or per-session basis. The service chaining logic is implemented as an XML script, and as such, standard IT tools and limited telecom knowledge are required to modify or customize the script. In addition, by using an XML-based approach, the Service Controller can download and execute user-specific or operator-specific scripts on a per-user or per-session basis, which is an important factor if a more open service delivery environment is being considered (e.g., where some of the application logic is hosted by 3rd-party ASPs or enterprise customers).

Conclusions

In this paper, we have described two different service architectures that have been used within HP's IMS Experience Center to implement an instant group communication application. The first approach is closely aligned with the 3GPP IMS architecture, where the core application logic was deployed as an application server function within the IMS network. In contrast, the second approach is more aligned with the OMA Open Service Environment (OSE), where network functions are exposed as web services and where the core application logic resides within a web environment.

Through this work, we can conclude that both approaches represent valid approaches to service delivery within an IMS environment. Indeed, while there are many differences between the two approaches, the two approaches also have much in common.

- Both approaches deliver a multi-modal and multimedia experience, as per the original user scenario description, and both approaches have been successfully implemented.
- Both implementations rely on a horizontal service delivery approach, where the core application logic acts as a "session orchestrator", coordinating the actions of application-independent service enablers.
- In both cases, application development was achieved using IT-standard development tools (in one case, using java development tools; in the other case, using a web services environment).
- Both projects have highlighted the importance of user interface aspects, and the importance of offering a seamless user experience across multiple services, multiple devices and multiple networks.

At the same time, as highlighted previously, each approach also brings its own unique set of benefits and challenges. The choice of which implementation model to adopt for a given service will depend on many factors, including the levels of flexibility, openness and performance that are required, the capabilities of the target end devices, and the functionality that is exposed by the network service enablers on their protocol and web service interfaces.

The choice of implementation model will often depend on the business model being adopted for a given service. The Application Server approach is likely to be used for applications that are deployed within the network service layer, while an approach based on the Service Chaining model is more likely to appeal to 3rd-party Application Service Providers (ASPs) who host applications outside of the network domain and who may wish to combine network service enablers with service enablers coming from other domains.

For developers and vendors of service enablers, the challenge will be to ensure that the same functionality is made available on both the protocol interface and web service interface, so that each service enabler can be used with both models. As such, the service enablement elements that populate a service delivery environment should not impose a specific service implementation model, but should be flexible enough to be used within multiple such models.

In conclusion, a layered horizontal approach to service delivery, within an IMS environment, can greatly help to reduce the cost of deploying and delivering new IP multimedia services, by enabling the re-use of common service enabler elements across multiple services. However, the adoption of such an architecture must also allow multiple service implementation models to co-exist. Through the IMS environment within the HP OpenCall Experience Center, HP is taking a hands-on approach to validate different IMS service implementation models and to understand the impact of these models on the architecture and the elements of the IMS service delivery environment.

Acknowledgements

Thanks to my colleagues at Hewlett-Packard, and in particular to Mark Gullett, Olivier Bertin, Paul Burke, David Isaacson and Sven Krause for their help and input on this paper.

Author Information

John O'Connell is based in Grenoble, France where he works for Hewlett-Packard's Software business, as chief architect for IMS within the OpenCall CTO Office. The OpenCall group is responsible for HP's telecom signaling, media, service and mobile management products. John joined Hewlett-Packard in 1986. He has been working in the network services area for over 10 years, initially in the IN domain, and more recently, in VoIP and IMS domains. During that time, he has been an active member of industry groups such as IN Forum and VASA Forum.

References

1. IP Multimedia Subsystem (IMS), 3G Partnership Program (3GPP), http://www.3gpp.org/.
2. IETF Internet Draft Message Session Relay Protocol (MSRP), http://www.ietf.org/internet-drafts/draft-ietf-simple-message- sessions-19.txt.
3. HP OpenCall Experience Center, www.hp.com/go/ims/.
4. OMA Group List Management Server (GLMS)/XML Document Management Server (XDMS), Open Mobile Alliance, http://www.openmobilealliance.org/.
5. OMA Service Environment (OSE), Open Mobile Alliance, OMA-Service_Environment-V1_0-20040907-A, http://www.openmobilealliance.org/.
6. HP Service Delivery Platform, http://www.hp.com/go/sdp/ .

VT