

# Value-Added Services in the Converged Network

*Yves De Serres and Lawrence Hegarty, Telebec*

## ABSTRACT

Now that the local market is open to competition, incumbent local exchange carriers as well as competitive LECs must not only follow the technology curve, but find a way to ride the wave of network convergence and offer innovative value-added services if they want to maintain their market share. Over the past decade, the intelligent network has been used to support value-added services. New APIs are being designed for the creation of value-added services spanning multiple domains (wireline/wireless, voice/data, public/enterprise). These APIs will not only ease the development of value-added services, but will also open service creation to third parties.

## INTRODUCTION

What is the key to success for service providers, in an environment where the boundaries between telecommunications, computing, and entertainment worlds are rapidly vanishing?

The successful service provider will:

- Deploy an efficient and broadly defined network infrastructure
- Offer, and continuously evolve, a vast portfolio of value-added, customizable, enhanced services
- Build the right business model to deliver these services profitably

In themselves, these objectives are not new. What grants them a new flavor is the increasing rate of technological advancement that is currently feeding the advent of the all-IP network. The Internet Protocol is touted as the grand unifier of voice and data networks into the so-called converged network.

Also emphasizing the importance of the above objectives are:

- The highly competitive environment fueled by deregulation
- The abundance of bandwidth generated by breakthroughs in fiber optics

Competition has brought service rates down, a trend that is likely to continue. As a result, it is

more and more difficult for service providers, especially the smaller ones, to compete on price.

Service differentiation will gain in importance as a weapon against the threat of competition. And ways to differentiate will abound due to the possibilities offered by the converged network. Indeed, it is easy to foresee that a unified network infrastructure coupled with high-bandwidth access technologies will promote the appearance of new telecom services and applications that will span multiple network domains (wireline and wireless, voice and data, public and private). While the true nature of these future services remains elusive, a few of their basic attributes may be listed:

- Future enhanced services will be immune to media boundaries.
- Subscribers will be able to customize enhanced services to their personal tastes and requirements.

This article first offers a broad view of the technology and economic factors driving the current wave of network convergence, and also shows how service providers, especially smaller ones, can exploit this convergence to become service integrators. The second part of the article is devoted to a new approach to service creation in converged networks.

We complement the above introductory comments by discussing the nature of the fundamental paradigm shift currently sweeping the telecom industry. Packet switching is rapidly overtaking circuit switching as the prime mechanism to support telecommunications as a whole (voice, data, video). The resulting "converged" network, most likely IP-based, will support services transcending domain boundaries (wireline vs. wireless, public vs. enterprise). As a byproduct of this convergence, the telecom infrastructure that used to be, and still is to a large extent, based on proprietary software and protocols is gradually evolving toward an open infrastructure based on open interfaces. This evolution has a dramatic impact on the service creation environment in particular.

The concept of the intelligent network (IN) was developed in the mid-1980s, standardized in the late 1980s, and implemented in the early

1990s. With the IN infrastructure, service providers are equipped with mechanisms allowing them to create, test, deploy, and support value-added *voice* services. We provide a high-level description of the IN model and infrastructure. The IN infrastructure has been deployed by virtually all large telcos as well as many smaller competitive local exchange carriers (CLECs), competitive access providers (CAPs), and wireless service providers (WSPs). While many success stories are attributed to IN, it is generally recognized that IN never fully achieved its promises.

Many standards bodies are defining plans to evolve the IN concept into the converged network. For many critics, however, these plans will only serve to lengthen the useful life of the IN infrastructure deployed at high cost over the last few years. These same critics advocate that the converged network provides a fertile ground for the development and support of enhanced services. The claim is that in evolving from a closed proprietary environment to a new environment based on open software and interfaces, the telecom industry will experience an explosion of new services much like the explosion of new applications that resulted from the transition from the mainframe to the PC in the computer industry. New application programming interfaces (APIs) supporting that claim are described later.

## THE TELECOM SHIFT

To better appreciate the challenge of deploying a service creation infrastructure in the converged network, we first draw a global picture of the fast-changing telecommunications landscape.

### NETWORK MULTIPLICATION

It seems that not so long ago there was a single network, the public switched telephone network (PSTN). Shortly after the advent of the mainframe computer, the need to interconnect remotely located computers appeared, a need which resulted in a new type of network, the data network.

Then the PC very rapidly became an indispensable tool in the enterprise domain where they are interconnected by local area networks (LANs). These LANs are themselves interconnected via corporate data networks. These private networks can communicate through the public data backbone. The aggregate of all these data networks became known as the Internet.

In the early 1990s, the World Wide Web made the Internet part of the general public's everyday life. Browsers provided a new mechanism for information sharing, information seeking, and information broadcasting. This led to a dramatic increase in the level of data traffic flowing in the public data domain. If observers disagree on exactly when it will happen, they all agree that the level of data traffic is soon to exceed the level of voice traffic.

Despite all the hype it creates, the Internet has to share the limelight with cellular communications. Wireless communications experienced widespread popularity in Europe due largely to Global System for Mobile Communications (GSM) technology. In North America, many WSPs exploit the latest digital technology for

wireless communications to compete with the incumbents that have been using analog technology since the early 1980s and are now upgrading to the digital infrastructure. Making less noise, the cable operators are upgrading their network. They already offer wideband access to the Internet and will soon offer telephony on top of their traditional TV broadcasting service.

This brief summary of the recent evolution of the telecommunications industry illustrates that the number of networks and the number of services have multiplied. While these networks may very well share a common transmission medium (e.g., fiber), they remain disjoint at the switching level. They are also very disjoint at the service level, a typical example of this being the multiplication of voicemail boxes that many business subscribers have to manage.

### THE AGE OF CONVERGENCE

Circuit switching is the basic transport mechanism<sup>1</sup> for voice in the PSTN, while packet switching was developed as the transport mechanism for data transmission. Each switching paradigm is particularly well suited to the type of communications it was devised to support.

The idea of using packet switching as a transport mechanism for voice (and/or video) is not new. The idea was considered very shortly after the advent of packet switching some 25 years ago. It was established, at least in principle, that packet switching could also support real-time communications.

However, at the time, the required technology was not available; nor were the telecom economics conducive to a migration toward a single switching mechanism for both voice and data. Recent technology advances and a new economic background are driving the convergence of the communications, information, and entertainment industries into a single business.

This wave of convergence will have profound implications for service providers. But before addressing how service providers can best ride the wave of convergence, we discuss a few of the drivers of this convergence that were alluded to above.

### TECHNOLOGY DRIVERS OF CONVERGENCE

Both integrated services digital network (ISDN) and asynchronous transfer mode (ATM) technologies were supposed to achieve convergence of voice and data networks. Neither fully succeeded, although each technology found an area of applicability: ISDN as an integrated access technology; ATM as an integrated transport mechanism in the backbone network.

The technology that is now touted as the future grand unifier is IP, the Internet Protocol. IP emerged as a leading candidate because of its prevalence in public and private data networks, its simplicity, and its key role as an enabler of the Internet as we know it today. IP is also pushed to the forefront by the recent and rapid improvement of real-time communications over data networks as exemplified by voice over IP (VoIP) technology.

Using IP as the communication mechanism for real-time applications like voice is not an easy task. By nature, IP networks offer *best-effort* transmission of packets. Real-time communica-

*The idea of using packet switching as a transport mechanism for voice (and/or video) is not new. The idea was considered very shortly after the advent of packet switching some 25 years ago. It was established, at least in principle, that packet switching could also support real-time communications.*

<sup>1</sup> The reader should distinguish transmission mechanism from transmission medium (copper, fiber, radio).

*Open APIs have greatly matured in the CTI environment over the past few years and have now started to move into the public domain. It is hoped that such an open telecommunications environment will attract application developers, just as the opening of the computing environment did.*

tions requires quality of service (QoS) guarantees that IP networks were not devised to provide. Research groups and standards bodies are working intensely on the definition of QoS mechanisms in IP networks.

Because of advances in digital signal processing (DSP) and the availability of interworking gateways between circuit-switched and packet-switched networks, IP is currently used in the public domain as an economical transmission mechanism for voice. This transmission efficiency is further increased by more robust core technologies like dense wavelength-division multiplexing (DWDM).

In the enterprise domain, convergence has been discussed for several years already and is referred to as computer-telephony integration (CTI). The infrastructure, centered on a CTI server connected to the corporate LAN and private branch exchange (PBX), allows the deployment of business-critical applications such as context-specific distribution of customer calls. In the CTI environment, the main driver of convergence is the increasing popularity of telephony APIs allowing the linking of corporate data and voice networks at the application layer. These developments in the enterprise domain are rapidly crossing the boundary with the public domain.

Above all, convergence is about services that can span circuit-switched networks (PSTN and cellular) and IP networks (public and private). In that respect, recent advances in distributed computing are fundamental drivers of convergence. Exploiting the capabilities of object-oriented programming, the new distributed processing technology provides an environment for service definition that hides the specifics of the underlying hardware infrastructure.

Finally, the legacy telecommunications hardware, based on mainframe and proprietary protocols and interfaces, is being replaced more and more by open-system client-server platforms. Open APIs have greatly matured in the CTI environment over the past few years and have now started to move into the public domain. It is hoped that such an open telecommunications environment will attract application developers, just as the opening of the computing environment did.

#### ECONOMIC AND REGULATORY DRIVERS OF CONVERGENCE

A few years ago, American and Canadian regulators concluded that competition in the local market would be in the best interest of the general public and thus removed previous restrictions on entry into the local market. In doing so, they followed up on previous regulations that opened the long-distance voice telephone service to competition.

For subscribers, the most noticeable effect of deregulation has been the continuous decrease of long-distance rates.

For service providers, competition coupled with the increasing availability of transmission bandwidth has created an environment where transmission capacity and basic services (either voice or access to the Internet) are becoming commodities. In a commodity market, profit margins are small and only large providers of the

commodity can remain profitable because of the high volume of the commodity they handle. This is one reason behind the recent wave of mergers and acquisitions between major national and international carriers.

In this context, traditional carriers have adopted strategies to diversify their service portfolio through alliances with cable television companies, Internet service providers, and wireless service providers. While these strategies will probably be profitable, the added value they generate is technology-based. In a highly competitive environment, this technology-based added value is unlikely to be a key differentiating factor simply because any new technology is likely to be deployed by most service providers in order to keep pace with the competition.

Successful service providers will exploit the capabilities offered by the converged network to define and market services that will enhance the communication experience of individual subscribers and increase the profit margin of corporate subscribers. These services will be the differentiating characteristics of competing service providers. This need to differentiate will thus fuel the convergence of networks.

Initially, service providers will probably use bundling of existing services as a differentiating characteristic. However, this may not be enough since new service providers will be quick to exploit the converged infrastructure to offer powerful integrated (voice and data) services. To do so, these service providers will migrate their current vertically oriented organization into an horizontally oriented model.

#### THE NEW NETWORK AND BUSINESS MODEL

Today, basic telecommunications services and the networks that support them display a vertically oriented organization. The networks (PSTN, cellular, data) share the same transmission media (e.g., fiber) at the physical layer.

As we move up to the network layer, we find that networks are essentially disjoint:

- The PSTN interconnects circuit switches.
- Cellular networks interconnect mobile exchanges.
- Data networks interconnect routers and ATM switches.

These networks are managed individually even when they belong to the same service provider.

Finally moving up to the application layer, we find that existing services are often "hard-coded" for a specific type of network or transport. This approach leads to a multiplication of the processes required to provision, bill, and manage these services.

One of the primary objectives of convergence is to provide a common, unified, and flexible control environment that can support multiple types of services over multiple types of transport media. As a result of convergence, networks will shift from a vertically oriented structure to a horizontally oriented model.

Globally, the horizontally oriented converged network has a three-layer representation, as in Fig. 1.

The network layer encompasses the access network, the backbone network, and the switching network. This layer provides connectivity. At

this layer we find the multiple transport media and switching mechanisms used today.

The network services layer provides service and session control. It bridges the application and network layers, allowing enhanced services to be defined at the application layer without consideration of the specific network technology used to support these enhanced services.

The application layer, which corresponds to the application layer of the open systems interconnection (OSI) stack, is where enhanced services are defined. Based on the shift toward an open architecture, developers of telecom applications will exploit open APIs supporting a rich flow of information between the application and network services layers. These APIs will open service creation to a large pool of application developers.

## THE IN PARADIGM

The three-layer network model described in the previous section clearly shows the separation between the service logic and the underlying network supporting this service. This separation is the defining attribute of the IN paradigm.<sup>2</sup> In this section we briefly describe how the IN paradigm is implemented in the PSTN.

### FIRST STEPS TOWARD IN

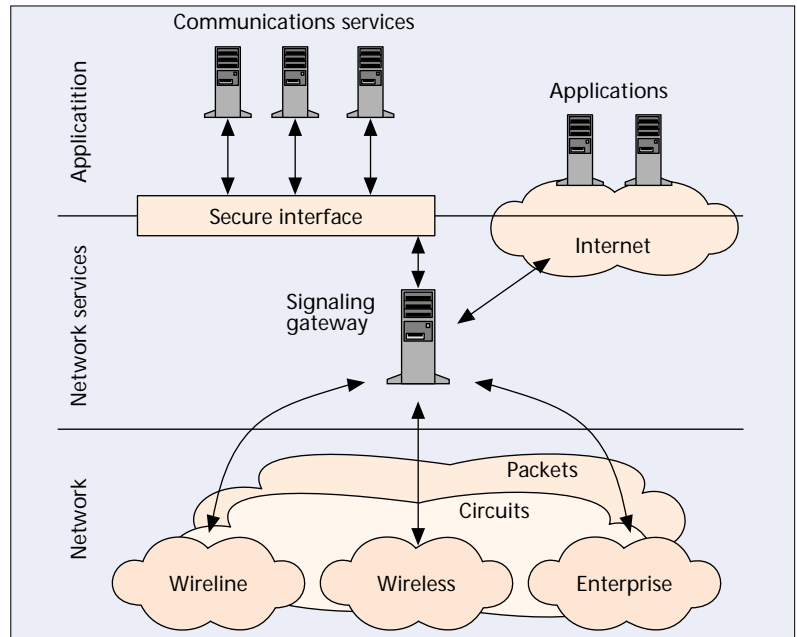
Two main events led to the conception of the IN paradigm:

- The advent of the electronic stored-program switch (1965)
- The migration from inband signaling to out-of-band signaling that resulted in the gradual deployment, through the 1980s, of the common channel signaling network based on the Signaling System 7 (SS7) protocol

The SS7 protocol is a layered packet switching protocol. The bottom three layers of SS7, the message transfer part layers, provide connectivity services. Two main protocols occupy higher layers of the SS7 stack, the Integrated Services Digital Network User Part (ISUP) and Transaction Capabilities Application Part (TCAP) protocols. ISUP provides call control functions and manages the setting up, routing, and tearing down of calls through messaging between network switches. TCAP was designed specifically to allow switches to communicate with general-purpose computers.

Because stored-program switches are computers, their basic functionality, switching calls, can be extended, through programming, to support calling services. This capability, combined with SS7, led to a category of services collectively referred to as custom local area signaling services (CLASS). CLASS services exploit the capability of transmitting calling party information to support such services as call blocking and caller ID. These services are switch-based services because they are supported by a piece of code residing in each and every switch. Despite the new functionality and added revenue achieved by switch-based services, their development cycle was very long and their implementation was under the control of equipment manufacturers.

The 800 number service is the first service that exploited the TCAP functionality. In order to route an 800 call, the switch where the call



■ Figure 1. A three-layer representation of the converged network.

originates issues a request to a centralized database of 800 numbers. This database stores, for each 800 number, information like:

- The actual NPA-NXX-XXXX number where the call should be routed
- The identification code of the interexchange carrier to be used
- The number to be billed for the call

This information is returned to the switch for it to properly route the call to its final destination. The basic translation service provided by the 800 database (managed by its front-end processor) can be extended to a more sophisticated routing mechanism by introducing context-sensitive conditions in the translation process (time of day, location of originating switch).

### THE STANDARDIZATION PHASE

Between 1985 and 1990, both the International Telecommunication Union (ITU) and Bellcore (now Telcordia Technologies) standardized the IN paradigm. The Bellcore standard is referred to as AIN for advanced IN.

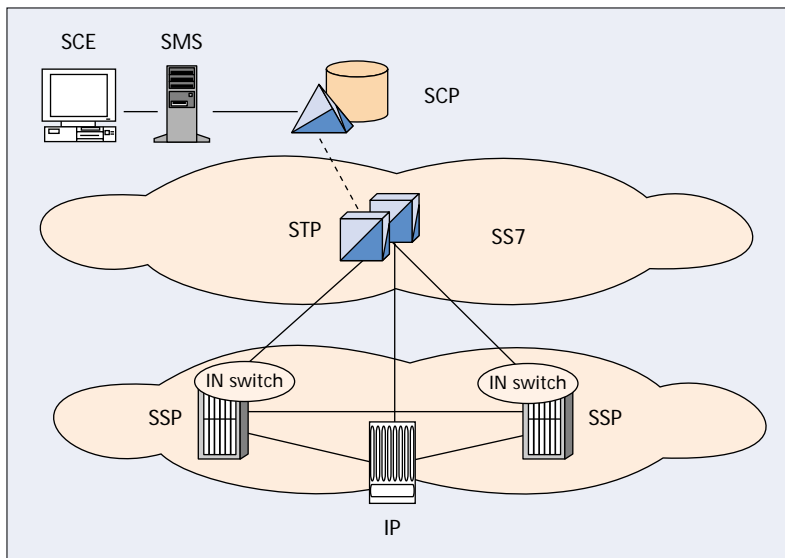
The main objectives of the standardization effort were:

- To provide a unified framework for the implementation of the IN paradigm
- To promote the development of value-added services in a multivendor environment
- To accelerate the development of customized service offerings
- To ease the management of cross-service interactions
- To migrate service development from equipment manufacturers to carriers

These standards define the IN model which is composed of a number of functional entities. Some of these functional entities may be collocated on the same platform or distributed on different platforms. We provide, in the following subsections, a brief description of these functional entities (Fig. 2).

<sup>2</sup> Intelligent Network Forum Web site: <http://www.inf.org>





■ Figure 2. *IN architecture.*

The basic call model is a state machine decomposing in-switch call processing into atomic phases.<sup>3</sup> The basic call model specifies so-called points-in-call where processing may be interrupted and additional service-specific instructions obtained from off-switch databases.

The service switching function (SSF) embodies the basic call model and the ability to communicate with remote databases. The SSF is a software package (feature) that can be added to the basic functionality of a circuit switch. This software package is often referred to as the IN software (or IN feature) in the switch. A switch upgraded with the IN software is called a service switching point (SSP) in the IN model.

The SCP is the central repository of the service logic (service control functionality). SCPs are robust computers acting as front-end servers to service databases where service data is stored (service data functionality). The SCP receives service requests from SSPs, executes the service logic, accesses the service database, and returns information used by SSPs to continue call processing. It should be pointed out that communication between SSP and SCP may occur at more than one point in call processing depending on the specific nature of the service associated with the call. It should also be pointed out that a SCP can host multiple services.

The SMS is attached to the SCP and performs management, provisioning, and maintenance of services supported by the SCP.

The SCE provides a set of basic tools (e.g., a graphical user interface) to assist in the development and testing of value-added services. The SCE offers a set of so-called service-independent building blocks (SIBBs) that can be combined to define specific services. Typical SIBBs providing elementary functionality are *Create call leg to number* and *Provide announcement*.

The intelligent peripheral is used to provide service-specific announcements and to collect digits (e.g., PIN numbers). IP is basically an interactive voice response platform.

The SS7 network is an integral part of the IN because it provides signaling interconnection

between platforms hosting IN functionality. The SS7 network is a packet network operating parallel to the PSTN. The central element of the SS7 network is the signaling transfer point (STP). STPs are routers of SS7 messages and are interconnected into a hierarchy of signaling networks.

#### THE DEPLOYMENT PHASE

Starting in 1992, incumbent LECs (ILECs) began to deploy the IN infrastructure in order to reap the benefits of value-added services. The following list provides a few examples of services deployed by service providers:

- Originating/terminating call screening
- Credit card calling
- Malicious call identification
- Selective call forward on busy or no answer
- Abbreviated dialing
- Follow me service
- Automatic alternate billing

This deployment continues to this day with the introduction of new value-added services. IN standardization also continues.

Despite many successes, IN deployment has suffered from:

- Incomplete adherence to the IN standard by switch manufacturers
- Very high cost of the IN infrastructure
- Complexity of the service creation process
- Complexity of service integration across wireline and wireless domains
- Complexity of integration with provisioning, customer care, and billing systems

#### THE FUTURE OF IN

Over the past few years, PC-based products have appeared that offer IN capabilities. These products use standard interfaces, open hardware platforms, and third-party software, and cost a few orders of magnitude less than the traditional IN infrastructure. Moreover, these platforms are more suited to the new environment where the boundaries between voice, data, and video services are continually blurred.

Large ILECs that have deployed an IN infrastructure in the recent past will be reluctant to retire this equipment in its early stage of depreciation. This situation is fully recognized by standards bodies that are actively working on the definition of interworking protocols that will allow the IN SCP in particular to maintain a key role in the converged network.

Moreover, the evolution of IN will certainly benefit from advancements in middleware and mobile agent technologies ([2-4]). A comprehensive survey of the impact of Internet, Common Object Request Broker Architecture (CORBA), Telecommunications Information Network Architecture (TINA), and mobile agent technologies on the evolution of INs can be found in [5].

#### THE PUBLIC SERVICE IP NETWORK

The specification of open APIs has gained momentum recently as a way to adapt the IN paradigm to the converged network. Telephony APIs have become popular in the enterprise domain where they ease the development and deployment of telephony applications. Examples of enterprise

<sup>3</sup> The basic call model will not be described in detail in this article. An introduction to the basic call model can be found in [1].

domain APIs are Microsoft Telephony API (TAPI) and Java Telephony API (JTAPI).

APIs are designed to hide as much as possible of the specifics of the underlying infrastructure in order to help the application developer concentrate on service logic and service attributes. Extending the API approach from the enterprise domain into the public domain has obvious appeal for service providers.

Public domain APIs will be key enablers of network convergence, promoting the development of value-added services spanning multiple domains (voice/data, wireline/wireless, public/private). These APIs will also provide a secure interface to network resources, thus contributing to the opening of the public network. It is expected that such an opening will entice application developers to create new value-added services.

In this article we provide high-level descriptions of two APIs that have recently gained some level of market recognition: Parlay and JAIN.

### THE PARLAY API

The Parlay group<sup>4</sup> was formed in April 1998 to produce an API allowing control of network capabilities. The original members were BT, Utlicom, Microsoft, Nortel Networks, and Siemens. In the beginning of 1999 the group was expanded to include AT&T, Cegetel, Cisco, Ericsson, IBM, and Lucent. In mid-2000, the Parlay group was incorporated into a non-profit organization and opened its activities to membership of any company wishing to participate in the evolution of the Parlay API.

The first version of the specification was released in December 1998. The group also produced a demonstration that showed how the API could be used by entities outside the network to produce services for customers using capabilities inside the network. As of November 1999, version 1.2 of the specification has been released. The second phase of the API is scheduled to be completed in December 1999. This phase will focus on the convergence of different networks, in particular, IP, PSTN, and cellular networks.

**Parlay Goals** — The goal of the Parlay group is to create a vendor, platform and technology-independent means to allow public access to telecommunications and data networks. The security of the network will be maintained, and as much of the underlying infrastructure detail as possible will be hidden from the user of the API.

A standard API provides vendor independence by hiding vendor-specific details. For example, a properly implemented Parlay API would allow an application that makes use of the generic call control portion of the specification to work on any brand of circuit switch without any change to the application code. The API's technology independence allows calls to be routed transparently through both wireline and wireless networks.

The Parlay API is also platform- and technology-independent in that it is defined independent of any programming language, operating system, or hardware. It can be implemented in C++ with Windows NT on an Intel-based PC or in Java with UNIX on a DEC Alpha-based machine. This frees users from being forced to deal with any one particular vendor.

**Elements of the Parlay API** — The Parlay API is divided into two portions: framework interfaces and services interfaces. The framework portion provides the infrastructure to support the services. The services provide a full range of network capabilities.

**Framework Interfaces** — The framework interfaces are independent of any service. They provide the tools needed to use the different services:

- **Authentication:** Applications use the authentication framework to indicate to the Parlay API who they are. There are multiple techniques that can be used, including encryption and digital signatures. Authentication is then used so that the application and network can agree on a service contract.
- **Discovery:** Once the authentication phase is successfully passed, the application can then use the discovery framework to find the services it requires.
- **Event notification:** The event notification framework is used to let an application know when certain generic service-related events take place.
- **Integrity management:** The integrity management framework allows the application to know when problems occur with the Parlay API or a particular service.
- **Operations, administration, and maintenance (OA&M):** As of phase 1.2 of Parlay, OA&M functionality is limited to time and date queries.

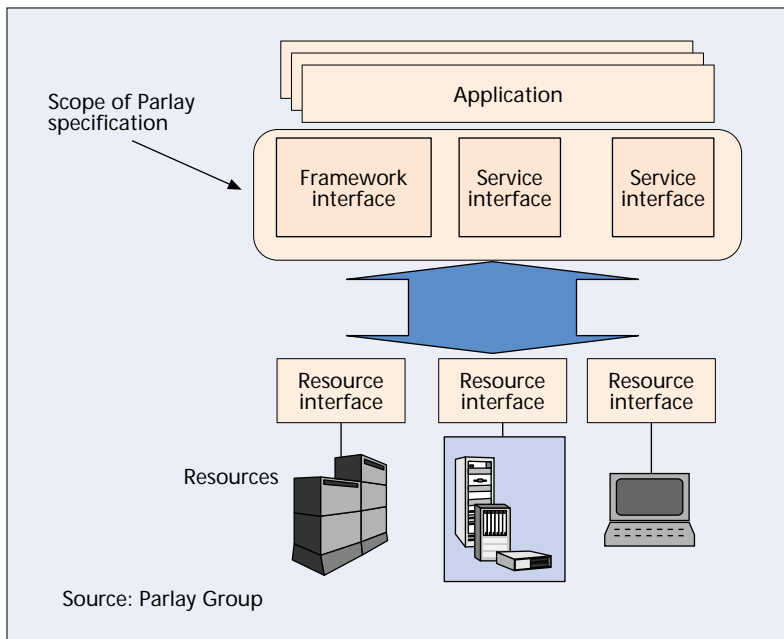
The framework services are crucial in that they are the portion of the Parlay API that allows secure access to the network. Third-party applications must pass through the authentication framework before getting access to network services. Framework services are the means by which network providers keep track of exactly who uses the network and how much they use the network.

**Services Interfaces** — The service interfaces are the vertical technology building blocks with which applications are created. They provide access to the network-level services needed to create higher user-level services.

- **Generic call control:** This service provides basic third-party call control functionality, and can handle both circuit- and packet-switched calls. This service is designed to be specialized into call control specifications (i.e., ISUP, H323, SIP).
- **INAP:** The INAP call control service specializes the generic call control service. It provides additional functionality such as the ability to request more dialed digits and the ability to supervise the call.
- **Generic messaging:** This service allows applications to send, store, and receive messages. The current version supports both e-mail and voice mail. The ability to control mailboxes, folders, and messages is provided.
- **User interaction:** Generic user interaction allows applications to interact with the end user.
- **Call user interaction:** This service extends

*Public domain APIs will be key enablers of network convergence, promoting the development of value-added services spanning multiple domains. These APIs will also provide a secure interface to network resources, thus contributing to the opening of the public network.*

<sup>4</sup> <http://www.parlay.org>



■ Figure 3. A logical view of the Parlay API architecture.

the generic user interaction and allows applications to interact with users participating in a call. It can play announcements and get feedback from call participants (via dial tone multifrequency, DTMF).

**Resource Interfaces** — Resource interfaces are the means by which a Parlay product interfaces with the network elements. The Parlay group does not specify these interfaces. It is likely that each piece of equipment in the network that interacts with a particular Parlay product will require an adapter. These adapters may be provided by the equipment manufacturer, built by third parties or included with a Parlay product. The availability of resource adapters may have a great impact on the ability of a Parlay product. The creation of resource adapters is a potentially difficult exercise. They may be prohibitively expensive for small service providers to create. A lack of public comprehensive technical information about a particular product may force reliance on the original manufacturer for creation of adapters. It is likely that resource adapter availability will be most critical in the early days of Parlay product availability. If the Parlay API is successful, increased demand should lead to increased availability of resource adapters. Early embracers of Parlay products must be careful to ensure the availability of the resource adapters they will require for the services they intend to create.

**Parlay Architecture** — Because Parlay is an API specification, it does not dictate any particular physical architecture. Applications may or may not reside on the same computing element as the Parlay code. The resource interface code can reside on the same CPU as the Parlay code, live on an intermediate computing element, or exist directly on the equipment it manages.

A logical architecture can be shown that sepa-

rates the different components of a Parlay product without regard for where they will exist at runtime.

Applications exist outside the network domain and interact with the network through the Parlay API. The Parlay API does not specify how communications between applications and the API will transpire. Middleware such as CORBA is a likely candidate for this task. The same middleware may serve as the communications bridge between the API and the resource interfaces.

The Parlay API will likely run on a machine called a Parlay gateway. This gateway will be connected to applications through an IP network. Parlay resource adapters are also likely to connect to the gateway through an IP network. The connection between resource adapters and the hardware they interface to will depend on how close to the hardware a particular adapter sits. In some cases equipment will be controlled indirectly, possibly through an SS7 protocol.

Figure 3 shows a likely Parlay-based product architecture.

At runtime, applications will communicate to the network by making API calls to the gateway (Fig. 4). These calls will sometimes result in interactions between the gateway and an adapter. If the adapter is protocol-related, it may inject packets into the SS7 network.

As of the fourth quarter of 1999, no Parlay products had reached the marketplace. Parlay solutions appeared toward the end of 2000. Despite being premature to judge the success of the Parlay API, it is safe to say that it offers network providers and third-party applications developers the highest-level approach to creation of value-added services in the converged network.

#### JAVA APIS FOR INTEGRATED NETWORKS

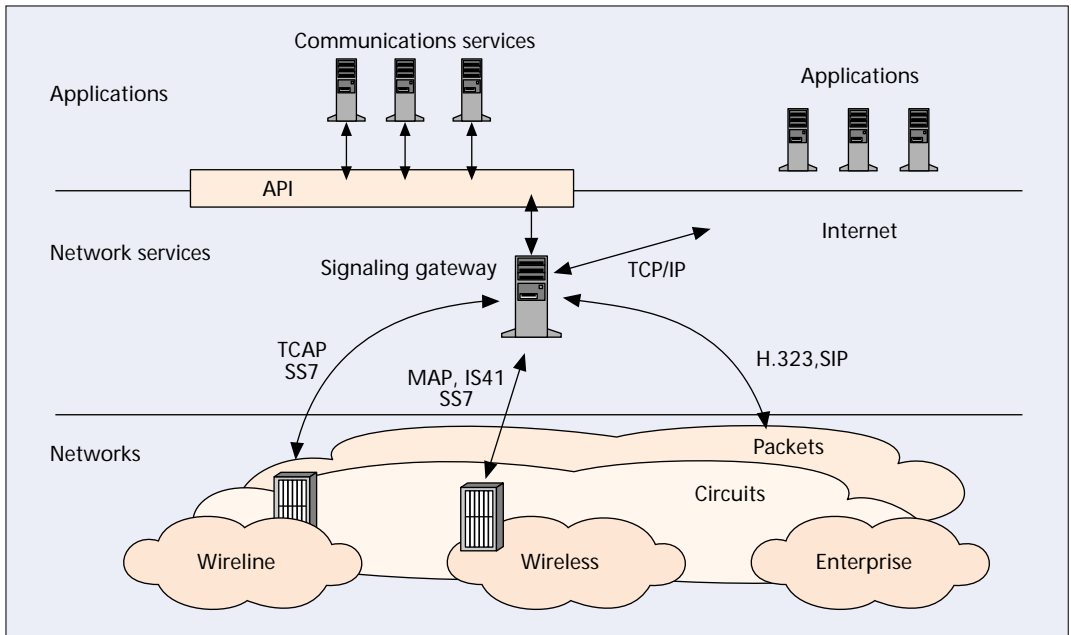
Java APIs for Integrated Networks (JAIN)<sup>5</sup> was born in June 1998 when Sun Microsystems and other companies in an alliance with Sun announced “the industry’s first Java technology-based solution for building and deploying state-of-the-art telecom services blending Intelligent Network (IN) and Internet technologies.” Originally the acronym stood for Java Advanced Intelligent Network (JAIN); it has since been changed to Java API for Integrated Networks since the first version was perhaps a misleading acronym.

JAIN comprises a software component library, a service creation environment (SCE), and a service logic execution environment (SLEE). The component library is an extension to the standard Java hierarchy.

**Protocols Expert Group (PEG)** — The software components are broken down into two different categories. The first category is the protocol part. The team working on these parts of JAIN is the PEG. Their work consists of APIs for several common protocols:

- Transaction Capabilities Application Part (TCAP)
- ISDN User Part (ISUP)
- Mobile Application Protocol (MAP)
- OA&M
- Media Gateway Control Protocol (MGCP)
- H.323 (ITU Recommendation)
- Session Initiation Protocol (SIP)

<sup>5</sup> [www.javasoft.com/products/jain](http://www.javasoft.com/products/jain)



■ Figure 4. The Parlay runtime architecture.

**Application Expert Group (AEG)** — The second JAIN component category is the application part. It includes higher-level APIs than the protocol APIs:

- **JAIN Call Control/JAIN Coordination and Transactions:** These APIs present a network and technology independent set of high level routines for manipulating calls. Using this API, calls can be transparently routed through different types of networks.
- **Secure Network Access:** JAIN will adopt the Parlay specification to provide a secure means to access network capabilities.
- **Service Creation and Service Logic Execution:** Defines the services that are guaranteed to be available in the SCE and SLEE. They are designed to make it possible to create services using SCE building blocks.
- **Connectivity Management:** An API to handle network policies and connections. A clearer explanation is not yet available from Sun at the time of this writing.

The SCE is a Java bean container that will allow service creators to build services using a GUI to dynamically link component beans.

The SLEE is a runtime environment for the JAIN services to work in. It will manage services, the resources they need, and their (secure) interaction with the network. It is somewhat analogous to a Parlay gateway.

The JAIN specification offers tremendous opportunities for network operators wishing to create value-added services. Java is a modern language with the benefits of built-in garbage collection and a small but concise syntax. If the Enterprise Java Bean technology lives up to expectations, it will add significant power to the JAIN API approach to service creation and execution.

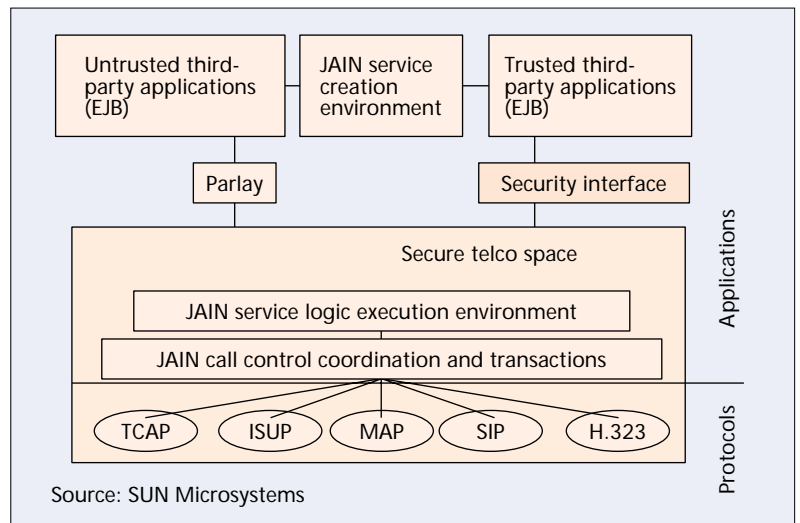
**Java Telephony API** — JTAPI is designed for CTI call control. It is used to create CTI applications such as call centers. The role of JTAPI is mostly for applications internal to the enterprise.

JTAPI is a competitor to Microsoft's TAPI.

Several areas of functionality overlap between JTAPI and JAIN. One of these areas is the JTAPI call control feature which is also used in JAIN. Much of the JTAPI functionality will eventually migrate into the broader JAIN specification.

**JAIN and Parlay** — JAIN and Parlay are seen as complementary solutions. Sun advocates the use of Java and JAIN to build Parlay services and gateways. Sun is adding a Parlay-compliant API hierarchy to JAIN. In some categories, such as security, the Parlay API may serve to provide all the JAIN functionality.

Figure 5 represents the JAIN runtime environment. It also shows how the Parlay API can fit into the picture to handle security issues for third-party service providers. It also highlights the fact that JAIN services have a full range of communications protocols available to them.



■ Figure 5. The JAIN application space.



Service providers are all looking for the killer application (or service). Maybe this is an elusive search that should be replaced by the development of a large portfolio of value-added services.

## CONCLUSION

The recent (ongoing) development of open APIs like JAIN and Parlay aim at simplifying the creation, testing, and support of value-added services in the converged network. Actually these APIs will likely be key drivers of this convergence since they should ease the creation of services spanning multiple domains (wireline/wireless, voice/data, public/enterprise). Service providers are all looking for the killer application (or service). Maybe this is an elusive search that should be replaced by the development of a large portfolio of value-added services subscribers can combine and customize.

## REFERENCES

- [1] "Intelligent Networks: Towards Implementation in Fixed and Mobile Networks," *IEEE Commun. Mag.*, Feb. 1992.
- [2] J. P. Redich, M. Suzuki, and S. Weinstein, "Distributed Object Technology for Networking," *IEEE Commun. Mag.*, Oct. 1998, pp. 100–11.
- [3] A. A. Lazar, "Programming Telecommunication Networks," *IEEE Network*, Sept./Oct. 1997, pp. 8–18.
- [4] M. Breugst and T. Magedanz, "Mobile Agents: Enabling Technology for Active Intelligent Network Implementation," *IEEE Network*, May/June 1998, pp. 53–60.
- [5] T. Magedanz, "Intelligent Network Evolution," tutorial presented at TINA '99 Conf.; <http://www.tinac.com>.

## ADDITIONAL READING

- [1] The HTRC Group, "The Future of IP Services," <http://www.htrcgroup.com>

- [2] A. Chadran, "The Bitter AIN," *Telephony*, Dec. 1998.
- [3] C. Wolter, "IN in IP Networks," *SoundingBoard Mag.*, Mar. 1999.
- [4] G. L. Andresen, "Serving Up the Open Network," *X-Change Mag.*, Mar. 1999.
- [5] C. Gbaguidi *et al.*, "Integration of Internet and Telecommunications: An Architecture for Hybrid Services," *IEEE JSAC*, vol. 17, no. 9, Sept. 1999.
- [6] C. Low, "Integrating Communication Services," *IEEE Commun. Mag.*, June 1997.
- [7] G. Vanecek *et al.*, "Enabling Hybrid Services in Emerging Data Networks," *IEEE Commun. Mag.*, July 1999.
- [8] P. Lambert, "Converged Creation Platforms," *X-Change Mag.*, Dec. 1999.
- [9] I. Foley, "How IP Will Change Your Service Paradigm," *America's Network Mag.*, Nov. 1999.
- [10] Y. Rabeau, "The World Telecom Industry: In the Eye of Change," *Forces Mag.*, no. 124, 1999.

## BIOGRAPHIES

YVES DE SERRES [M] ([ydeserres@telebec.qc.ca](mailto:ydeserres@telebec.qc.ca)) received his Ph.D. degree in electrical engineering from McGill University. He also holds a Master's degree in mathematics from Université de Montréal. His 20 years experience spans the academic environment (INRS — Telecommunications), the equipment manufacturer environment (Nortel Networks), and the service provider environment (Microcell). He joined Telebec, a subsidiary of Bell Canada, in 1999, where he is senior director, Systems and Services Integration.

LAWRENCE HEGARTY ([lah@jacaza.com](mailto:lah@jacaza.com)) received his Master's degree in computer science from Concordia University and his undergraduate degree in computer science and philosophy from Potsdam College. He started developing Java software with Sanga International, and has worked for Nortel Networks and contracted for companies such as eNet Global, Epsilon Data Services, and Telebec.