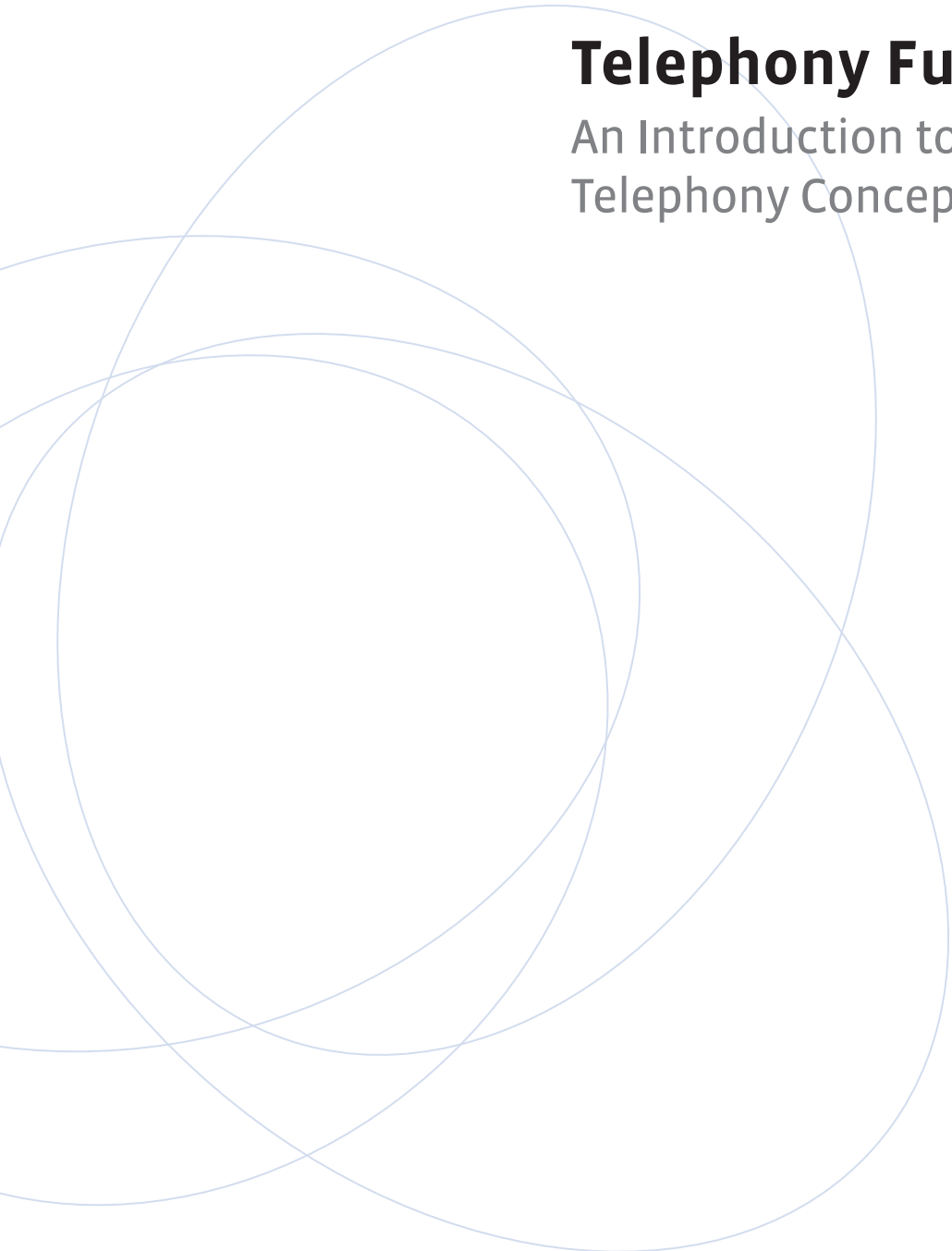


Telephony Fundamentals

An Introduction to Basic
Telephony Concepts



Executive Summary

As telecommunications has moved from proprietary to open, standards-based systems, advanced voice solutions have grown richer and more cost effective. Several basic telephony concepts are critical to working with these solutions: call control, media processing, in-band and out-of-band signaling, and local, dedicated first-party control versus shared, network-based third-party control. Once these concepts are understood, today's modular, converged, and increasingly web-centric communications technologies become easier to understand.

Table of Contents

Open Telephony Solutions	2
How Is Telephony Used?	2
Why Progress Was Rapid	2
Telephony Basics.....	2
What Is Signaling?	2
In-Band and Out-of-Band Signaling.....	3
CTI.....	4
First-Party and Third-Party Call Control.....	5
Application Programming Interfaces	6
Modular Media Processing Hardware	7
Web-Centric Voice Applications	8
VoiceXML and SALT	8
A Wealth of Options.....	8
For More Information.....	8

Open Telephony Solutions

Open telephony solutions emerged with the introduction of commercially-available computer telephony technology in the mid-1980s, but the predominance of expensive, closed proprietary systems, which were used to handle functions, such as voice processing and computer-based fax, held back their wide adoption.

Today open, standards-based telephony technologies are widely available and are rapidly becoming converged and web-centric.

How Is Telephony Used?

Basic telephony concepts are the foundation for advanced voice processing solutions. Here are just a few current examples of such solutions:

- Call centers
- Self-service Interactive Voice Response (IVR) systems
- Unified messaging
- Application media servers
- Voice over IP (VoIP)

Why Progress Was Rapid

Several factors combined to simplify open telephony systems significantly and accelerate their deployment:

- International standards for interconnecting telephone and computer systems were defined, such as the Computer-Supported Telephony Application (CSTA) from the European Computer Manufacturers Association (ECMA) for linking computers to telephone systems.
- Application Programming Interface (API) specifications, which provide a set of software calls and routines that can be used by an application to access communications services, became widely accepted.
- Voice processing technologies continued to add advanced features and increased port density at attractive prices.
- The deregulation of public networks encouraged new and innovative service and equipment providers to enter the market segment.

Today a new group of developers are beginning to build voice applications using a web application infrastructure with standards-based interface languages such as Voice eXtensible Markup Language (VoiceXML) and Speech Application Language Tags (SALT). A basic knowledge of how telephony works allows these developers to leverage

their current skills to create converged voice and data applications as easily as they create data applications for the PC and Internet.

In addition, microprocessor technology continues to advance to the point where processing no longer has to be performed on specialized silicon, but can take place on the host processor of an off-the-shelf computer. This host-based media processing technology stands to help significantly reduce the total cost of ownership of telephony equipment.

Telephony Basics

Public and private telephone systems provide real-time information paths between two or more parties. The wireline public system is generally referred to as the PSTN (Public Switched Telephone Network) and private systems are created with PBX (Private Branch eXchange) or KTS (Key Telephone System) switching technologies.

Traditionally, these public and private information paths have taken the form of voice connections, originally through hardwired analog circuitry but later through an increasingly broad range of technologies, such as radio transmission, digital signal encoding, and fiber. Over time, these transmission paths also came to be used for non-voice applications, such as fax and data transmission.

When equipped with the proper software and hardware, computers today send and receive every kind of information that passes through the telephone network. Computers now can do all of the following:

- Act as fax machines
- Interact with human speakers through voice recognition and synthesis
- Provide gateways between different kinds of networks, such as the PSTN and networks enabled with VoIP

It is this intersection, with the general-purpose computing platform serving as the interface point, which makes telephony so valuable.

What Is Signaling?

The telephone network is a widely distributed system of intelligent switching nodes. Signaling is the process by which nodes communicate to establish and tear down (conclude) calls so that two or more parties can communicate via terminal equipment (such as a phone acting as an endpoint) and the network.

Among the most well-known schemes for terminal-equipment-to-network signaling is the Dual Tone Multi-Frequency (DTMF) protocol, under which the terminal equipment generates simultaneous pairs of tones to represent each dialed digit. Over time, more protocols were developed to facilitate communications between switches and endpoints (known as “line connections”) and switch-to-switch communications (known as “trunk connections”). Today’s signaling protocols take the form of tones and digital messages, such as Channel Associated Signaling (CAS), Integrated Services Digital Network (ISDN), Signaling System 7 (SS7), and Session Initiation Protocol (SIP).

Two functions of a voice application describe how it interacts with the rest of the telephone network:

- **Call control** – An application must control how calls are established, reconfigured, and “torn down” (the telephony term for concluding a call). Session control in a data network is similar. Picking up the telephone handset, pressing dialing digits, and listening for the tones signaling the successful completion of a call are examples of “human” call control.
- **Media processing** – An application must send and receive information through the call endpoint interface, generating and receiving the appropriate information formats, such as fax, voice, tones, or data. The human equivalent of media processing is speaking and listening to the other party once the call is established.

A basic voice mail system provides a simple illustration of how call control and media processing interact. The system answers incoming calls, presents a greeting, and then records the caller’s message. Such a system consists primarily of media processing functions, with call control functions limited to detecting a ring, answering the call, and hanging up after the message has been taken.

More sophisticated applications, such as automated attendant, require more complex call control functions (call transfer, the ability to initiate calls, and conferencing) and advanced media processing, such as speech recognition and speech synthesis.

Call center applications typically utilize advanced call control and media processing functions. These include special call control functions to monitor calls as they pass through holding queues on their way to their ultimate destinations, and comprehensive media processing functions in an IVR system, which allows some callers to

complete their transactions without ever speaking to a human service representative.

In-Band and Out-of-Band Signaling

An accurate and reliable signaling connection between telephone and computer systems is essential to successful voice applications.

Signaling on circuit-switched networks can take place “in-band,” that is, on the same channel with voice communications (known as the “talk path”), or “out-of-band” (that is, through some communications channel other than the talk path). In today’s telephone network, terminal equipment or line connection signaling is generally in-band (except for ISDN devices), while signaling between telephone switches or trunk connections is often done out-of-band for security and performance reasons.

A very reliable way to implement signaling between telephone systems is to use out-of-band signaling, which creates a direct, message-based digital information link between the telephone switch and the computer-based application platform. This approach is much more accurate than in-band signaling, under which the application must attempt to generate and recognize widely varying and ambiguous analog signals in the call’s talk path.

Out-of-band signaling is available in several forms:

- The data channel (D-channel) associated with basic and primary rate interfaces (BRI and PRI, respectively) in ISDN lines
- The proprietary digital signaling between PBXs and digital telephone sets
- The switch-to-switch signaling protocol called SS7 used in public and large private telephone networks
- The Computer Telephone Integration (CTI) links available for many modern PBXs and some public exchange switches
- H.323 and SIP protocols used in VoIP networks

For an illustration of out-of-band signaling using SS7 technology, see the call routing application in Figure 1. In this example, Party A calls a special “virtual” telephone number associated with an endpoint, such as an 800 number, triggering a number translation request from the switch through the SS7 signaling network to the application system. The application system responds with a number translation giving the actual telephone number of Party B. The switch then applies standard routing rules

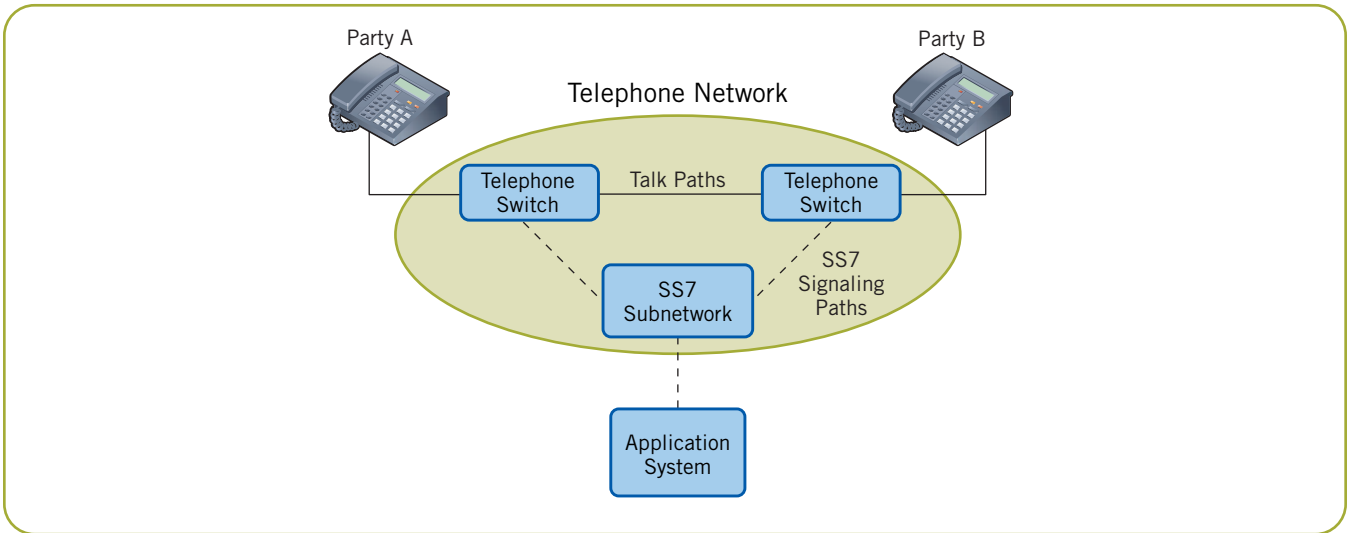


Figure 1. Out-of-Band Signaling Using SS7 Technology

to make a connection between Parties A and B. The talk paths are dedicated to carrying the voice signal only.

Signaling protocols on VoIP networks (for example, SIP, H.323, MGCP/Megaco) evolved from out-of-band, message-based protocols of the PSTN. CTI links, now offered on most modern PBXs, offer a signaling mechanism through which a computer system can receive consolidated signaling for groups of telephone extensions.

Telephone networks have existed for decades. As a result, a wide variety of standard and proprietary protocols exist. This has made it difficult for systems developers to effectively connect to telephone networks.

The good news is that today's telephony and web-oriented development environments abstract application developers from many of the underlying complexities of

the telephone network. As long as developers understand the capabilities and limitations of common signaling protocols, they can design sophisticated applications on a high level.

CTI

CTI links are also used for out-of-band signaling, but on a scale more suitable for the smaller and relatively simpler environment of a customer premises PBX or a single public telephone exchange switch rather than in the core of the PSTN.

CTI links often offer a broader range of call control functions than commercial customer-premises SS7 services, including call initiation and hang-up as well as call routing. CTI links can operate using either a proprietary protocol or a standard protocol such as CSTA.

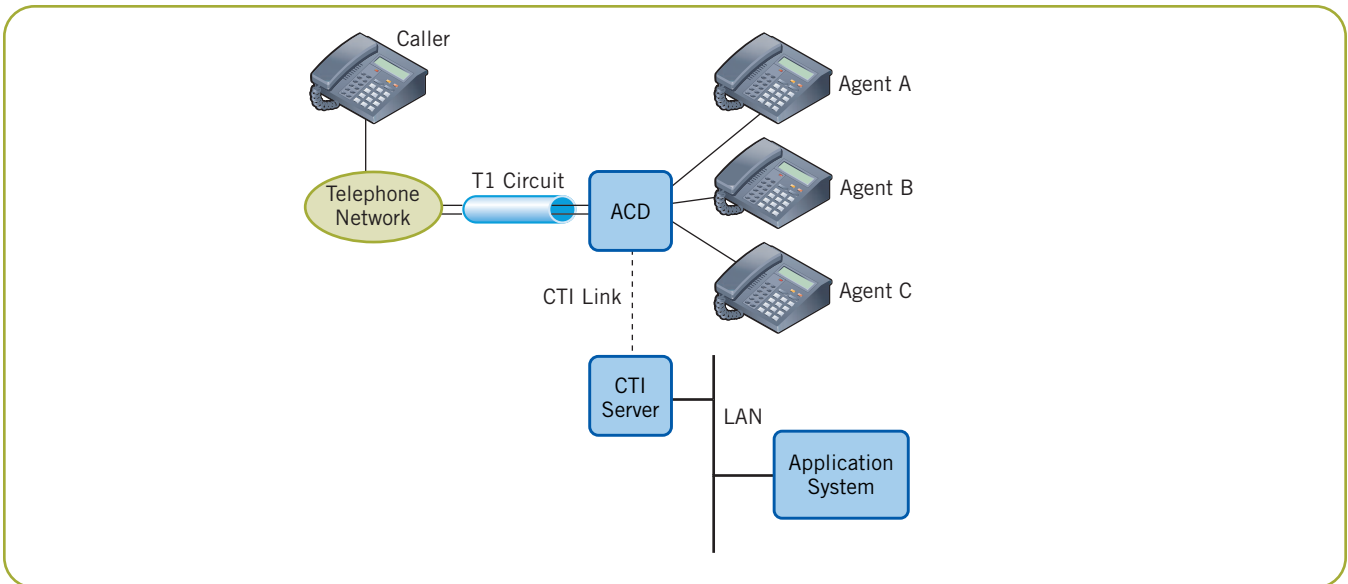


Figure 2. Out-of-Band Signaling Using a CTI Link

Figure 2 shows how CTI links can be used in a call center. The caller dials a number associated with the call center's T1 circuit. The public network delivers the call with associated calling-party and dialed-number information. When the call arrives, the Automatic Call Distributor (ACD) sends a message, including this information, to the CTI server via the CTI link. The CTI server passes the message to the application system, which responds with a routing instruction. The CTI server passes this routing instruction to the ACD via the CTI link, and the ACD completes the call to the designated call center agent.

Because they provide access to shared resources, both SS7-based connections and CTI links typically terminate in a server rather than a specific application computer. This allows multiple applications to influence calls flowing through a common telephone domain, and provides greater flexibility for the computer systems on which these applications can be installed.

First-Party and Third-Party Call Control

First-party and third-party call control are ways of defining how an application handles a call.

In first-party call control, an application performs call control functions locally. Here are some examples:

- A computer equipped with a telephony voice board and connected to a telephone line senses a ring signal, answers the call, and initiates a voice mail application to greet the caller and record a message.
- An incoming call is routed to a desktop, where an application on the desktop computer uses the Caller ID information accompanying the call to look up data about the caller in a database on the PC. The application displays the information as the call rings on the desktop phone or headset. See Figure 3. Calls can also be re-routed after a set number of rings to a pager, another device, or another location.
- A user accesses a personal call list on a desktop PC and tells the application to initiate a call to someone on the call list.

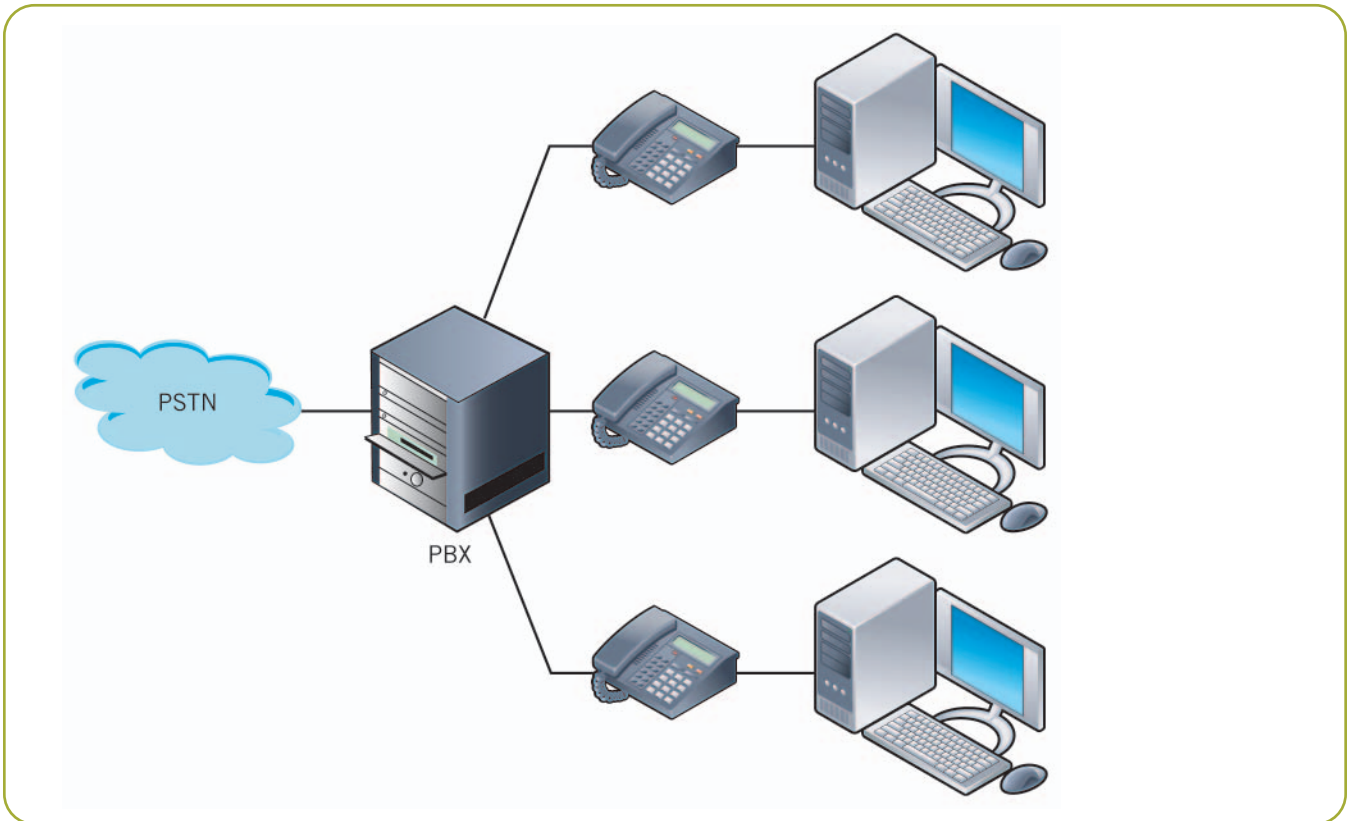


Figure 3. First-Party Call Control Example

In third-party call control, an application on a server or switching device handles calls for a group of users. Call control is handled on the network and not locally, and media resources are shared. Here are two examples:

- A server-based application monitoring several users' telephone lines, without an actual physical connection to each of those lines, sends an incoming call to one of the telephones on the network based on pre-determined criteria. The application can also initiate a conference among several users on the network, or can transfer calls from desktop to desktop.
- A server-based application uses the Caller ID information that comes in with a call to look up information on the caller in a network database and send that information along with the call to be displayed as the call rings at a desktop. See Figure 4.

Third-party call control usually implies out-of-band signaling, since there is no direct connection between the computer system running the application and the telephone line being controlled.

Third-party call control is far more efficient and less expensive. Since only a small number of desktop phones in an organization are in use at any given time, the

application on the network can manage a pool of incoming lines and provide service to a large number of users on demand.

Application Programming Interfaces

A telephony API is the mechanism through which application software manipulates telephony resources. APIs are necessary for both call control and media processing functions.

In the spring of 1993, Microsoft® and Dialogic jointly introduced the Telephony Application Programming Interface (TAPI), which standardized the interfaces to telephony applications on the Windows® operating system and removed the difficulties inherent in addressing a variety of switches individually. A similar API was the Telephony Services API (TSAPI) developed by AT&T and Novell.

Although both TAPI and TSAPI were important steps toward standardization, both APIs were oriented toward first-party use on the desktop, that is, for local, non-shared resources.

Today the industry is working toward standardized XML-based APIs, such as CCXML and ECMA323 for

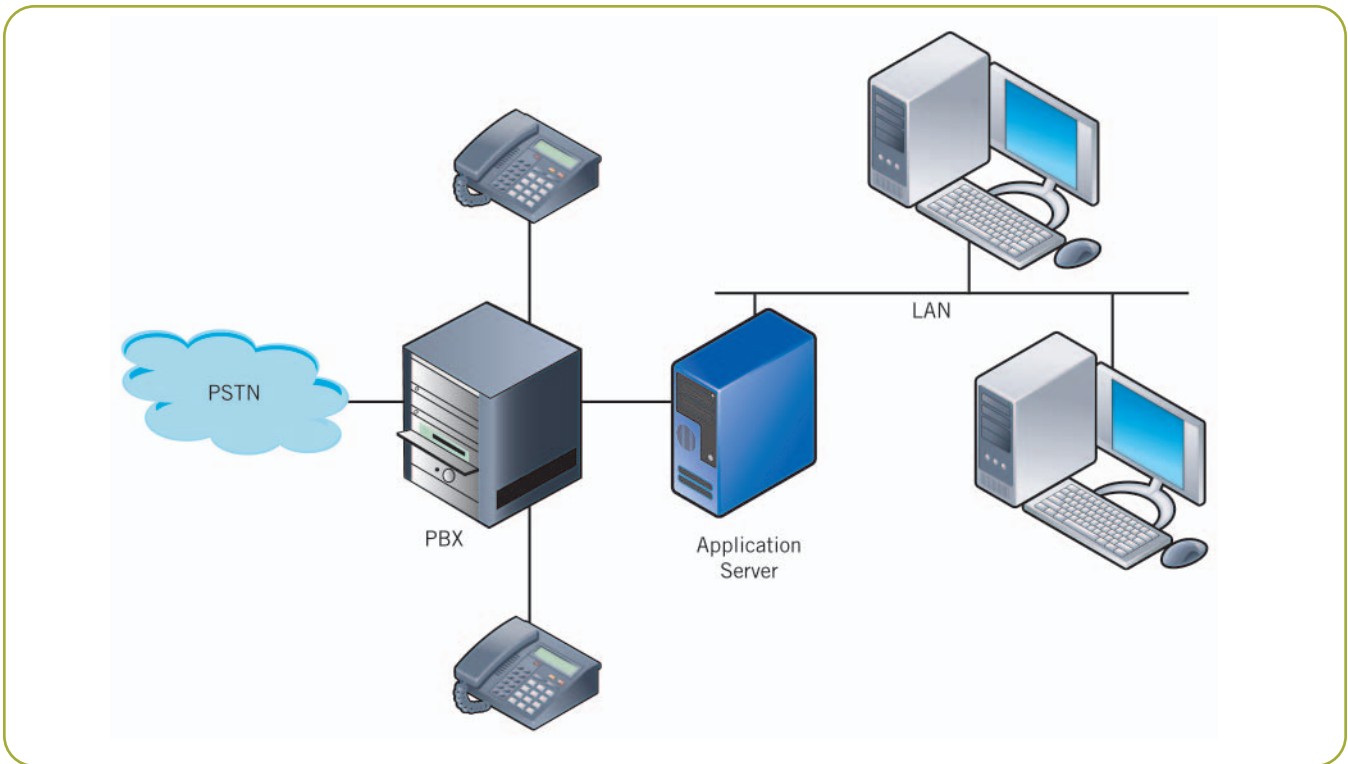


Figure 4. Third-Party Call Control Example

speech and call control, and MRCP for media resource management. And as telephony moves toward a web-centric model, XML-based APIs, such as WSDL and SOAP, are gaining importance.

Modular Media Processing Hardware

Traditional, proprietary systems used dedicated hardware. A first step toward modularizing telephony systems came with the development of voice processing boards implemented in standard computing form factors such as Industry Standard Architecture (ISA) and Peripheral Component Interconnect (PCI). In addition, standards were created for pooling these resources using a common bus architecture. Examples include SCbus, Mult-Vendor Integration Protocol (MVIP), and CT Bus (H.100/H.110).

Voice processing boards utilize digital signal processors (DSPs) as the engines for voice processing. As forecasted by Moore's Law, microprocessor technology has advanced to the point where many of these tasks can now be implemented on the host processor of an off-the-shelf computer instead. Dialogic refers to this technology as host media processing.

Today, host media processing software is making great strides in reducing costs. Although DSP price-performance has improved over the years, it has not kept pace with the price-performance of standard silicon such as the Intel Pentium processor. Dialogic® Host Media Processing (HMP) software moves media processing from the specialized DSP to a standard host processor, such as the Pentium chip. This move offers key advantages:

- **Lower cost of inventory and startup** — Initial capital investment is smaller
- **Lower development costs** — Development systems do not require specialized hardware
- **Lower deployment costs** — Software is less expensive to install and configure than hardware
- **Lower sparing costs** — Hardware can be used for multiple functions
- **Lower maintenance costs** — Maintenance is easier and less training is needed when system configurations are standardized

Web-Centric Voice Applications

With the advent of the Internet over the past ten years, a new architecture, with new software tools, has emerged to support the development and deployment of Web applications. “Web services” refers to a specific aspect of web applications technology involving structured transactions, utilizing XML encoding, and flowing between web servers.

Increasingly, a web-centric model is being applied to voice applications. Communications solutions have traditionally been standalone systems that had little customized interaction with other applications in an enterprise. In contrast, a web-centric model provides important integration advantages. Data and communications systems can share physical infrastructure, development staff, and support resources. This capability is referred to as “Communications Web Services” (CWS), defined as the provisioning of communications services so that they can be controlled and managed by web-style applications.

VoiceXML and SALT

Just as the growth of the web was catalyzed by the creation of the HTML scripting language, the acceptance of standards for speech services is propelling the adoption of this technology in both traditional and web-based applications. Two emerging language standards, Voice eXtensible Markup Language (VoiceXML) and Speech Application Language Tags (SALT), enable developers to do the following:

- Write platform-independent applications that handle synthesized speech.
- Recognize spoken input and DTMF
- Record spoken input
- Allow telephony control

VoiceXML markup tags are specifically intended for defining speech user interfaces while SALT markup tags can define “multimodal” user interfaces involving both speech and a range of devices, such as a graphic display, mouse, keyboard, or pen. (Note that a recent extension of

VoiceXML called “X+V” is also intended to define multimodal user interfaces.) VoiceXML defines a complete standalone language with markup elements for defining a speech interface along with data and control flow. SALT defines a small set of tags for creating a speech interface within various markup environments such as HTML, Wireless Markup Language (WML), Synchronized Multimedia Integrated Language (SMIL), and others.

VoiceXML and SALT represent two different approaches to a speech markup language. Each provides open specifications and supports the industry preference for known tools and programming models.

A Wealth of Options

Open telephony solutions have grown from simple applications that used computing technology to enhance proprietary systems to flexible, diverse, and low-cost applications involving high-density, high-availability communications systems based on open telephony standards, and, increasingly, web-centric models.

Today telephony is at an important turning point. The basic elements of the technology have been developed and standardized. Dialogic is driving modularity and standards for telephony and communications, and anticipates similar benefits for solution providers and application users alike.

Dialogic’s unique telephony experience allows it to supply modular hardware and software building blocks for the communications infrastructure. Companies can work with these building blocks to put together converged and modular communications solutions.

For More Information

Dialogic® Products and Resources

For access to Dialogic® telecom product information and resources such as white papers, design guides, and applications notes, visit <http://www.dialogic.com>.

To learn more, visit our site on the World Wide Web at <http://www.dialogic.com>

Dialogic Corporation

9800 Cavendish Blvd., 5th floor
Montreal, Quebec
CANADA H4M 2V9

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH PRODUCTS OF DIALOGIC CORPORATION OR ITS SUBSIDIARIES ("DIALOGIC"). NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Dialogic may make changes to specifications, product descriptions, and plans at any time, without notice.

Dialogic is a registered trademark of Dialogic Corporation. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. Other names of actual companies and products mentioned herein are the trademarks of their respective owners. Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement their concepts or applications, which licenses may vary from country to country.

Copyright © 2007 Dialogic Corporation All rights reserved.

07/07 3146-05