CHAPTER 1

Introduction

XCOFFEE was not what the designers of the Internet had in mind when they connected computers at research sites across the United States under the auspice of DARPA. Nor was it an expected application when the nuclear scientists at CERN invented the Web to facilitate information sharing. *XCoffee* refers to a video frame-grabber, installed in 1991, that is connected to a camera focused on a coffee machine in the Trojan Room at the Computer Laboratory of the University of Cambridge in England. Its installation had a very simple motivation. The Laboratory had a coffee club that shared the use of the coffee machine. Club members often negotiated several flights of stairs only to find the coffee pot empty. Frustrated, a couple of club members rigged up a system—including the camera, the frame-grabber, a client program, and a server program—in a day or two so that, with *XCoffee*, Cambridge students and faculty members could check, from the convenience of their computers, whether there was enough coffee or whether a new pot needed to be brewed. Eventually, everyone with access to the Web, from anywhere in the world, could view the status of the coffee pot (`http://www.parkerinfo.com/coffee.htm`). At the time, *XCoffee* was viewed with more amusement than understanding, but it served as a good indicator of what was to come.

In this chapter, we briefly trace the trend in Internet technology that led to the vision and opportunity of a networked home. We describe the service gateway as the functional nucleus in such an environment, depict the challenges facing software vendors trying to enter this market, explain the pioneering work of the Java Embedded Server™ product, and introduce the Java™ technology-based solution specified by the Open Services Gateway Initiative (OSGi) for developing and deploying services for the home.

1.1 The Internet and the Networked Home

Since the early 1990s, the Internet has grown tremendously and its impact is now felt in almost every aspect of our lives. It has changed the way we obtain entertainment, communicate, and conduct commerce, and is becoming a household presence like the telephone and the television.

Because of the ubiquity of the Internet and the vast resources it makes available, a computer that is *not* connected to the Internet becomes less useful, and the features available locally become less relevant. In fact, many people own a general-purpose computer solely to access the Internet.

As the Internet matures, *services* are being offered in addition to contents. We shop, trade stocks, plan trips, and get news on the Internet, and we want to do these things not only sitting at a desk, but also on the road. Thanks to hardware that is getting more powerful and less expensive, many smart and connected devices such as mobile phones, pagers, and personal digital assistants have emerged on the market. As a result, a traditional computer cannot meet the requirement of staying connected while at the same time be portable, specialized, and convenient. People have begun to dub this trend the **post-PC era**.

As we bring computing technology to our daily lives, we find our homes full of promising opportunities. On one hand, broadband services such as digital subscriber line (DSL) and cable modem are widely available, a growing number of households own multiple personal computers (PCs), and all forms of home entertainment are becoming digitized. On the other hand, various device control networks, pioneered by CEBus, Echelon, and X10, have already been developed for use inside the home.

Cahners In-Stat reported that the number of cable modem subscribers in North America reached 1.8 million in 1999, and the number of broadband cable data subscribers worldwide will be 9.5 million by 2002. In fact, today more than 110 million homes in North America are within a short distance of a broadband coaxial cable line and 77 million homes have cable TV services. The research firm also predicts that the average number of connected nodes per home network will increase from 2.9 in 1999 to 5.0 by 2003 [1].

Enthusiasts predict fascinating applications: refrigerators ordering groceries automatically, TV programs delivered based on personal interests, merchandise offerings catering to consumer taste, optimally tuned climate control systems for comfort and energy conservation, not to mention microwave ovens that can be turned on from afar.

The market potential is huge. Cahners In-Stat estimates that the revenue for cable broadband services will increase from its current \$1 billion to \$4 billion (US) by 2002. Alliend Business Intelligence projects that home networking equipment market alone will reach \$2.4 billion by 2005, whereas Parks Associates estimates that the total value of the end user market will be more than $4.5 billion by 2004 [1].

Given the market size, it is not surprising that a large number of companies are investing millions of dollars in developing technologies, creating standards, and manufacturing novel products. Numerous field trials are also underway as a necessary step in understanding customer needs. Although the application for service gateways is in the pilot stage, some of the dreams have begun to take shape. For example, the TiVo service allows you to rate TV programs and record your favorites automatically through a recorder connected to the service provider over the phone line. The Echelon LonWorks control network allows you to control light switches, window blinds, and thermostats through a browser over the Internet (`http://demo.echelon.com`).

As another example, in the Danish capital of Copenhagen, a trial began in September 2000 in which each family received a futuristic Electrolux Screenfridge, a Tele Denmark 2-Mbit/second ADSL high-speed data connection, and an Ericsson WAP phone. For the next few months, the families had the opportunity to try out an array of applications—Internet e-mail, intrafamily messaging, Web access, online grocery shopping, weather information, and news—targeted to improving the quality of life at home [1].

Some of us will really enjoy a good successor of *XCoffee,* a coffee maker that not only sends us detailed status information about itself automatically, but also produces personalized coffee at just the time we need it.

1.2 The Service Gateway

Traditional service providers such as utilities, telephone, and cable TV companies all have their own dedicated wires into the home. However, in the envisioned networked home of the future, this configuration will soon become unmanageably complex for a diversified portfolio of services, such as home security, health monitoring, telephony, and audio/video media, each possibly using a different communication technology. It will also miss interesting opportunities for integration. As a result, a centralized device interfacing the external Internet and the internal device and appliance networks has been proposed (Figure 1.1). This device is called a **service gateway**.[1]

[1] This term is used interchangeably with **residential gateway** in this book, although it could be argued that a service gateway may operate in settings other than the residence, such as a retail outlet or an information kiosk. They refer to the same type of device in terms of what they do, however.

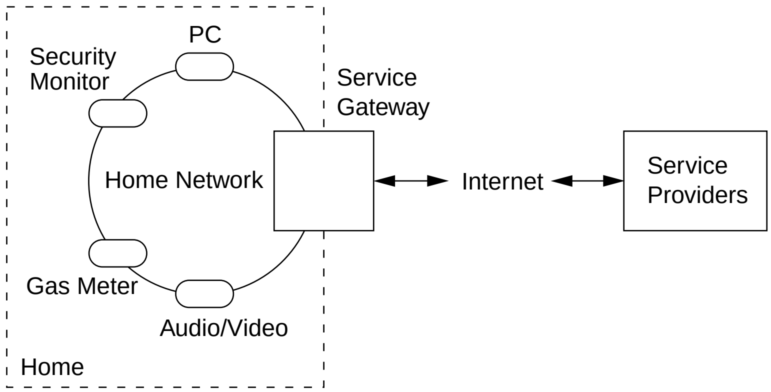

Figure 1.1 The role of a service gateway

A typical native platform of a service gateway consists of

- A processor
- Memory
- Persistent storage (disk or flash RAM)
- TCP/IP networking
- A device network (for example, a serial or parallel port)
- An operating system or real-time operating system

The familiar cable set-top box can be augmented with more “smarts” and transformed into a residential gateway. For example, Motorola’s DCT-5000+ set-top is equipped with a 300+-MHz MIPS processor, 14 MB of memory, an integrated cable modem, the Ethernet, Universal Serial Bus (USB), and IEEE 1394 interfaces. Its main application is to provide traditional video services and high-speed Internet access [2].

Other companies have designed service gateways from the ground up. For example, Ericsson’s e-box features a 100-MHz 486 CPU, 32 MB of memory, 24 MB of flash memory, a 10BaseT Ethernet interface, and a serial port. Its purpose is to provide e-services: Internet access, alarm and security, remote energy control and management, health care, e-commerce, and entertainment [3].

These are just a few examples of what is being tried out today. As costs decrease, more powerful devices can be expected to emerge on the market.

The residential gateway can participate in a wide range of home-based services:

- Home security, fire alarm, disaster alert, and emergency response
- Home-based health care, patient diagnosis, and child-care monitoring
- Energy management involving heating, ventilation, and air-conditioning

The benefits of delivering services to homes are twofold. For the consumer, her demands are met exactly when and how she needs them inside her home; intermediaries are eliminated so that she can hope for cheaper products and services. For merchants and service providers, they can serve a much more focused market more efficiently and can open new revenue streams with value-added services. The residential gateway can also leverage one service to benefit another. For example, it is possible for the home theater system to send a signal to lower the automated window blinds when a movie starts on a Sunday afternoon. A gateway that can authenticate a user via a cell phone can also let the user remotely open a door when a family member forgets to bring the key.

1.3 Challenges

Although optimism abounds, unique challenges are to be met on this new frontier. Two primary concerns are the large number of competing solutions and an uncharted new environment in which to develop and deploy applications for homes.

1.3.1 A Multitude of Competing Solutions

Although IP over T1 and the Ethernet are the winning formulas for wide area networks and local area networks (LANs), respectively, many solutions are vying to serve the home. Although we do not offer detailed treatment of these alternatives, we feel it is beneficial to present a high-level view of the technological landscape.

For physical media used to network the home, the consensus seems to be that the existing infrastructure should be leveraged and no new wires should be introduced. Consequently, we end up with the promising contenders presented in Table 1.1.

When it comes to connecting devices, there are many interfaces from which to choose. Table 1.2 summarizes some of the common device interfaces and their data rates.

Table 1.1 Physical Media within Homes, Their Data Rates, and Related Consortia

Physical Media	Data Rate	Associated Consortium
Infrared	9,600 bps–115 kbps; up to 4 Mbps	The Infrared Data Association (IrDA) `http://www.irda.org`
Phone line	1–10 Mbps	Home Phoneline Network Alliance (HomePNA) `http://www.homepna.org`
Radio frequency	1–2 Mbps	Home Radio Frequency Working Group (HomeRF) `http://www.homerf.org`
Power line	PowerPacket technology from Intellon at 14 Mbps	Home-Plug Powerline Alliance (HomePlug) `http://www.homeplug.org`
TV Cables	27 Mbps downstream, 500 Kbps–10 Mbps upstream	Cable Television Laboratories, Inc. (CableLabs) `http://www.cablelabs.com`

Table 1.2 Device Interfaces and Their Data Rates

Device Interface	Data Rate
Serial port	115 Kbps
Parallel port	115 KBps–3MBps
USB	12 Mbps
IDE	3.3–33 MBps
SCSI	5–160 MBps
IEEE 1394 (Firewire or iLink)	100–400 Mbps

Many comprehensive protocols are also out in the arena. These protocols are generally built on top of physical and data link layers, and focus more on the functionality provided by application layers in the OSI reference model. Some of them address a particular application domain—for instance, LonWorks, CEBus, BlueTooth, and Home Audio Video Interoperability (HAVi), whereas others attempt to provide a generic framework for enabling all devices—Jini™ connection technology and Universal Plug and Play (UPnP) are examples; these protocols are catalogued in Table 1.3.

Table 1.3 Other Prevailing Protocols

Protocol	Built On	Applications
LonWorks `http://www.echelon.com`	Power line, twisted pair, and so on.	Control network for devices such as sensors, switches, and instruments
CEBus (Consumer Electronics Bus) `http://www.cebus.org`	Power line, twisted pair, coaxial cable, radio frequency	Communication among residential consumer products
X10 `http://www.x10.org`	Power line	Communication and control among household electronic devices
BlueTooth `http://www.bluetooth.com`	2.4 GHz radio link	Sending and receiving voice and data within 10 m between portable devices wirelessly
HAVi (Home Audio Video Interoperability) `http://www.havi.org`	IEEE 1394	Allowing digital consumer electronics and home appliances to communicate with one another
Jini technology `http://www.sun.com/jini` `http://jini.org`	Java platform	Allowing a Jini technology-enabled device to join a federation and publish its service without prior setup. Others in the community can look up and discover the service, which is defined by a service interface and is represented as an object in the Java runtime environment. The object is exchanged using RMI.
Universal Plug and Play `http://www.upnp.org`	IP networks	Allowing devices and their capabilities to be discovered, utilized, and controlled. Devices are addressed by IP addresses; what services they provide and how they are controlled are described by XML.

1.3.2 A New Application Environment

The potpourri of standards provides pieces to the puzzle, whose completion, however, is anywhere but close. We have little experience creating and managing software applications in the residential market, which is made vividly clear by the following parody [4]:

This article first appeared in Salon.com, at `http://www.salon.com.` *An online version remains in the Salon archives. Reprinted with permission.*

Sunspots: Excerpts from a Diary of a Networked Future

By Mark Gimein

"[Sun Microsystems CEO Scott] McNealy gave several examples of the Net connected future: Light bulbs will be able to warn when they're about to expire, letting the factory automatically deliver a replacement. Vending machines will bill you automatically when you order a Coke with your cell phone. And the TV set-top box will be the nerve center of home networks that tie together dishwashers, thermostats, video cameras and everything else."

—From CNet News.com, Nov. 17

. . . . . .

Jan. 17, 2003
Help! The washing machine has crashed and will not give up my socks. When I try to open the door, the screen flashes "Error in scripting routine, line 18637." I see the socks spinning inside. Apparently I have put in a mismatched pair, and the machine doesn't seem to like that. I think I am not the only one who has had trouble. McNealy was on television last night, saying that we could reduce processor load by investing in clothes that we can "wash once, wear many times."

. . . . . .

June 9, 2003
I spent three hours today trying to get Sprint to reverse the charges for a 45-minute call to Turkey, but have had no success. I tried to explain that I did not call Turkey, but simply bought a Coke from the vending machine using my cell phone. The phone people say they understand, but it was the new Java-enabled bottle cap that actually made the call. The bottle caps report on a random sample of consumer purchasing behavior. But somehow a programming mistake had the bottle cap misdialing. Instead of calling the toll-free customer response number, it dialed into an international network.

. . . . . .

> Aug. 24, 2003
> Windows 2003 for Set-Top Boxes came out yesterday. I didn't want to spend $595 for an upgrade, but I don't think I really had a choice. It turns out that Windows 2000 was incompatible with my washing machine, and some of my clothes are still trapped inside. The new version promises 100 percent compatibility with major appliances. The trade-off is that it is not fully compatible with the embedded operating system in the toaster, but there might be a workaround. I'm told that in extremes I can control the toaster directly through Sun's Web site. Or maybe I can just do without English muffins for a while. The toaster has a slow microprocessor, and toasted muffins just don't seem all that important when you have to wrestle the computer to get them.
>
>

Humor aside, the story leaves few doubts that programming for devices of the future should not be taken lightly. Programming for a conventional computer deals with data that seldom have direct physical consequences; data integrity can be protected by means of backups, transactions, and other techniques. However, it becomes a little difficult to recover from a home appliance gone awry.

The setup and configuration of the device should be simple and automatic, whereas complicated appliances and services should be managed remotely by the service providers. It is well known that many people cannot program their VCRs. For them, managing smart appliances must be as simple as turning on the radio. Often, a service such as recording a TV program depends on the presence of another service, such as an electronic programming guide. Thus it must be possible for the residential gateway to detect the need for a service, discover its presence or absence, and ensure that it is available.

The Java Embedded Server product from Sun Microsystems is targeted to the residential gateway market and is designed to meet these challenges. By adding the Java Embedded Server software to a hardware product, device manufacturers can easily transform any broadband termination device, such as a DSL/cable modem or set-top box, into a residential gateway.

1.4 Java Embedded Server Technology

Since their inception, the Java programming language and the Java platform have been widely accepted as the way of programming for the Internet. Because the Java programming language is platform neutral, developers can write and test applications in a desktop environment such as the Solaris™ Operating Environment or Windows, then deploy them to the target device. An application written for one device can be directly deployed on another device as long as the other device provides a Java™ virtual machine. The Java runtime environment has the unique capability of loading code securely at run-time from the network. As a programming platform, the Java programming language protects developers from many well-known programming mistakes that have plagued C programmers. It shortens development cycles and boosts code quality.

Having realized the great potential of applications on smart connected devices, and the strengths of the Java platform in addressing this class of software, Sun Microsystems developed and released the first version of the Java Embedded Server product in October 1998. Its architecture represents a unified software programming interface that allows services to work together. (A more precise definition of service in the context of the OSGi architecture is given in the next chapter. Here it simply means a functional component.) A service, as a component, can be programmed to implement any protocol or perform any function in an insulated or cooperative manner. More specifically, the foundation of the Java Embedded Server product, a framework known as ServiceSpace, deals with the following issues that are important for provisioning services to the residential gateway:

- **Just-in-time service delivery.** The framework allows for a service to be downloaded over the network when the service is needed. The service may be used once and then discarded or it may be kept persistently on the residential gateway for a longer period.
- S**ervice updates and versioning.** The framework can be used to check quickly the version of a service that is running in a gateway and to update this service dynamically from a remote location. This is very useful for developers of software for embedded devices. These devices have traditionally been loaded with a static application environment. A critical application bug can be expensive to fix. Using the Java Embedded Server product, this is no longer a restriction. A newer version of the software can be loaded into the framework through the network.
- **Service discovery and dependency resolution.** To leverage components that others have developed, the framework provides a service discovery mechanism with which a downloaded component can consult a service registry in the framework to obtain and use an existing service. The framework also resolves dependency relations when one service depends on another to function.

Shortly after version 1.0 of the product was released, the cartoon in Figure 1.2 appeared, to the amazement of the members of the project.

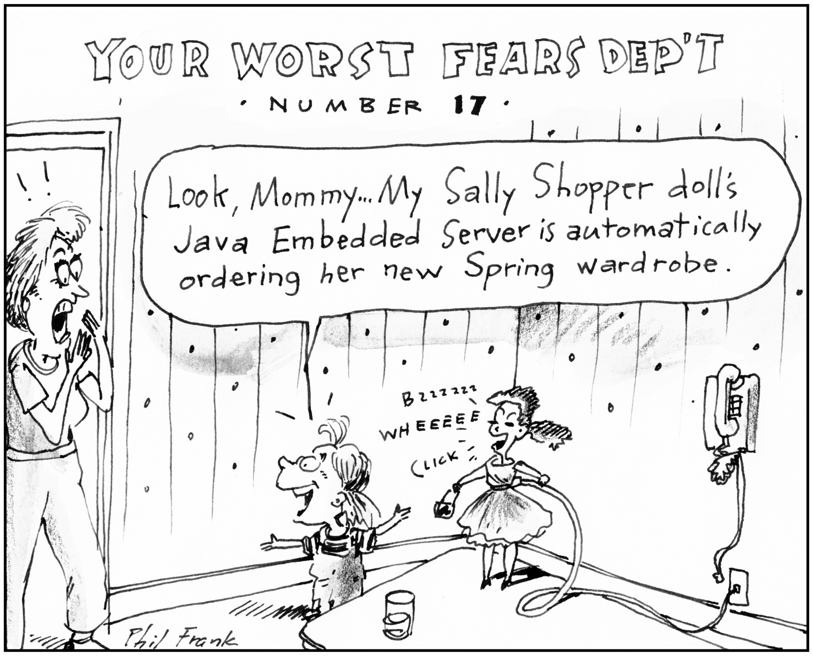

Figure 1.2 A misuse of the Java Embedded Server product. It seems to suggest that the general populace is quite ready to accept the prevalence of technology in their daily lives. (Illustration by Phil Frank.)

The Java Embedded Server product has gone through several revisions, including versions 1.0, 1.1, and 1.2. The current version is 2.0, released in August 2000. To simplify and expedite the development process for services, this version is bundled with a customized version of the Forte™ for Java™ Community Edition, an integrated development environment, as well as a number of prepackaged services. The Java Embedded Server version 2.0 software is an implementation of the Service Gateway Specification 1.0, released in May 2000 by the OSGi.

1.5 The Open Services Gateway Initiative

Sun Microsystems is a founding member of the OSGi consortium (`http://www.osgi.org`), an independent, nonprofit industry group that has defined an open standard for connecting future generations of networked consumer and small business devices to Internet services. The need for standardization is obvious:

- **Platform independence.** Different types of gateway devices usually have their own native platforms and various combinations of processors and operating systems. Such diversity is much more common for the embedded devices than for desktop computers. It would be infeasible for service developers to write applications and port them to many platforms with drastically different underlying characteristics, let alone have them work with one another seamlessly. This problem can be solved by leveraging the "write-once-run-everywhere" value proposition of the Java technology and by agreeing to a common standard for the gateway software.
- **Vendor independence.** An open specification focuses on defining application programming interfaces (APIs), making it possible for residential gateways manufactured by different vendors to host services written by different service providers that are provisioned by different gateway operators. The roles of various participants are discussed in the next section.
- **Future-Proof.** An open standard coupled with the dynamic upgradability of the Java technology means that a residential gateway hardware box has longer utility. It can remain the same while the user changes service providers or adds new services.
- **Integration.** An open standard enables coexistence and integration with multiple LAN and device access technologies, and can provide a software platform that can accommodate existing technologies so that, for example, a Jini™ technology-enabled printer, a HAVi-compatible audiovisual receiver, and a UPnP camera, can all interoperate via the standard platform.

The specifications that the OSGi consortium produces provide a common foundation for Internet service providers (ISPs), network operators, and equipment manufacturers to deliver a wide range of services via gateway servers running in the home or remote office.

The release of Java Embedded Server software version 1.0 predates and in fact catalyzed the formation of the OSGi. The product attracted several software companies, service gateway hardware vendors, and service providers. Sun Microsystems began working with IBM, Ericsson, and others to standardize the API to ensure interoperability of software solutions in the residential gateway market. In March 1999, 15 companies formed the OSGi consortium. The OSGi Java Expert Group was officially established, and one of the coauthors of this book, Li Gong, was elected by the group members and formally appointed by the OSGi Board of Directors as the first Java Expert Group Chair. Sun Microsystems contributed the specifications of Java Embedded Server software version 1.1 (the latest version of the product at the time) to the OSGi, which became the basis and a starting point for the specification work by the Java Expert Group. The group eventually produced the official OSGi Service Gateway Specification version 1.0, which was approved by the OSGi Board of Directors and released at Connections 2000, an event held in San Diego, California, in May 2000. Although it stemmed from the Java Embedded Server product and its ServiceSpace architecture, the OSGi specification has evolved the APIs and other features. As a result, Sun Microsystems has rebuilt the Java Embedded Server product from the ground up and released the OSGi-based version 2.0 of the product in August 2000.

The OSGi has attracted lots of attention and has been growing fast. As of this writing, more than 80 companies, including many notable device manufacturers, software developers, network operators, and service providers, have joined the OSGi.

1.6 Operational Model

The residential gateway market and the business model associated with it are still in the early stages. Probably the closest resembling practices in existence today are ISP and the cable TV companies. However, the extrapolation depicted in Figure 1.3 seems to be reasonable.

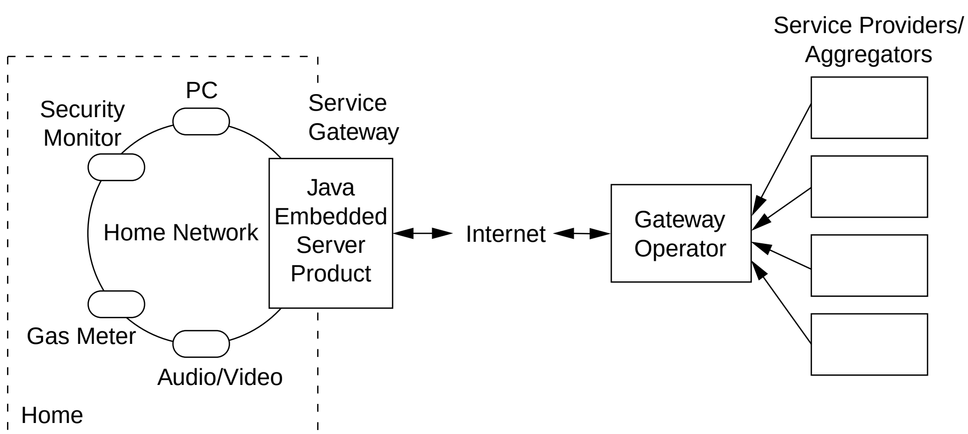

Figure 1.3 Operational model using the service gateway with the Java Embedded Server software.

For example, to avoid costly customer visits by technicians, an energy company may wish to deploy a piece of software that runs on home gateways in residence and reads gas meters remotely. The energy company would be the **service provider**. Additionally, the following players may also be involved:

- A company that manufactures the gateway box hardware
- Another company that supplies basic OSGi-based framework and core service software. (This portion of the software provides a generic infrastructure and services for deployment of other application components. In Figure 1.3, the Java Embedded Server product from Sun Microsystems is an example.)
- A **gateway operator** to install and manage the meter-reading software for its subscribers
- A **service aggregator**, because many services may be in demand at home, to integrate and package services from different service providers, then deliver them to the gateway operator.

One entity can serve multiple roles. A gateway hardware vendor may also supply the framework and other “horizontal” software; a gateway operator may also act as the aggregator.

Unlike a conventional computer, the gateway box usually does not need to have a display or input device such as a keyboard or a mouse. It should be maintenance-free from the home user’s point of view, and should be administered through the network by the gateway operator. The box should boot up as soon as it is powered up.

The hardware aspect of the service gateway and its associated business model is not discussed further in this book. Beginning with the next chapter, we focus on the software that runs the service gateways.