The unique aspect of IoT, compared to other network systems, of course, is the presence of a number of physical things and devices other than computing or data processing devices. Figure 15.1, adapted from one in Y.2060, shows the types of devices in the ITU-T model. The model views an IoT as functioning as a network of devices that are tightly coupled with things. Sensors and actuators interact with physical things in the environment. Data capturing devices read data from/write data to physical things via interaction with a data carrying device or a data carrier attached or associated in some way with a physical object.

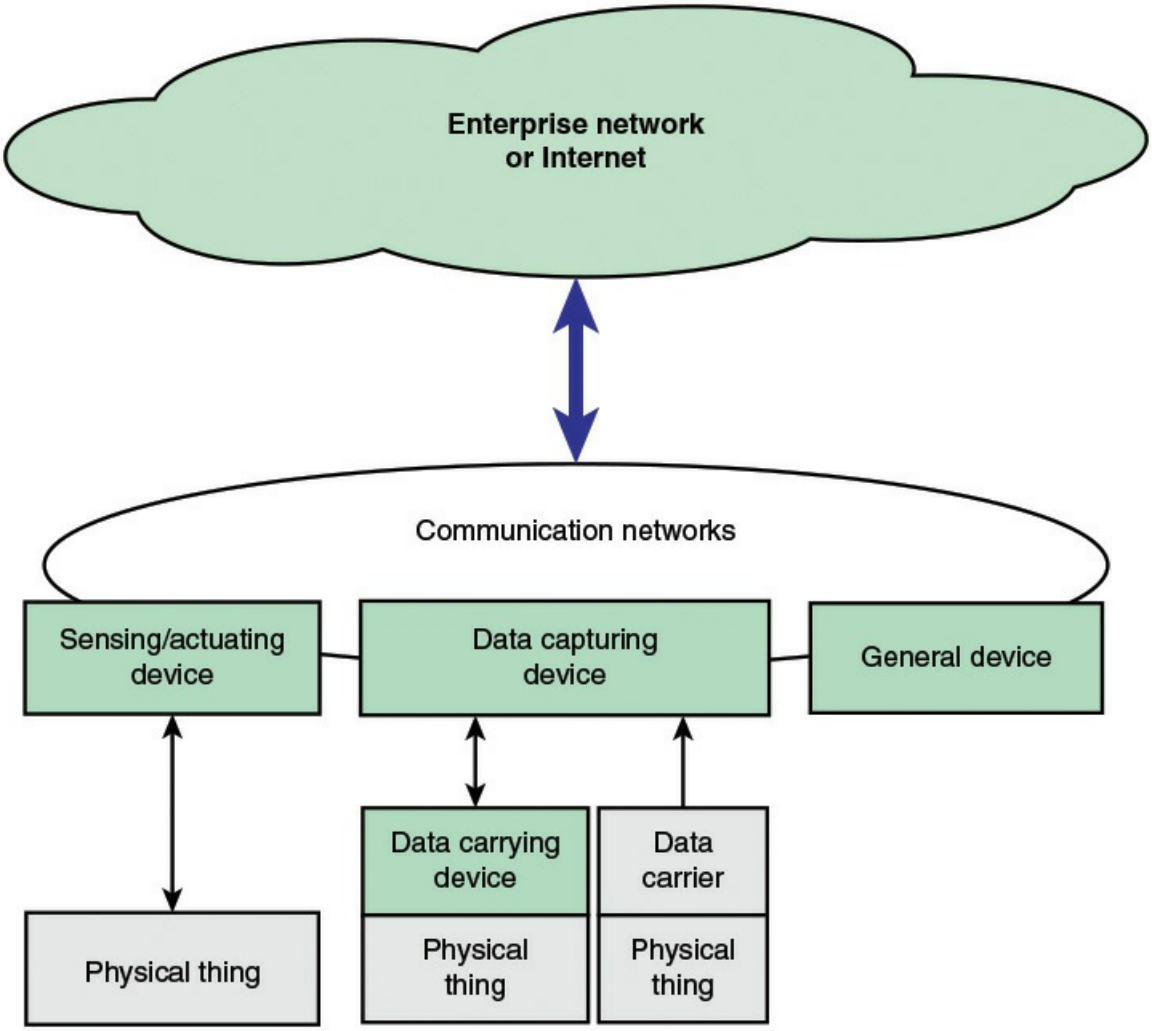

FIGURE 15.1 Types of Devices and Their Relationship with Physical Things

The model makes a distinction between data carrying devices and data carriers. A data-carrying device is a device in the Y.2060 sense. A device at minimum is capable of communication and may include other electronic capabilities. An example of a data-carrying device is an RFID tag. By contrast, a data carrier is an element attached to a physical thing for the purpose of identification of providing some other sort of information.

Y.2060 notes that technologies used for interaction between data-capturing devices and data-carrying devices or data carriers include radio frequency, infrared, optical, and galvanic driving.

Examples of each include the following:

- **Radio frequency:** An RFID tag is an example.
- **Infrared:** Infrared badges are in use in military, hospital, and other settings where the location and movement of personnel needs to be tracked. Examples include infrared reflective patches used by the military and battery-operated badges that emit identifying information. The latter can include a button that must be pressed so that the badge can be used as a means of passing through a portal, and a badge that automatically repeats the signal as a means of tracking personnel. Remote control devices used in the home or other settings to control electronic devices can also easily be incorporated into an IoT.
- **Optical:** Bar codes and QR codes are examples of identifying data carriers that can be read optically.
- **Galvanic driving:** An example of this is implanted medical devices that use the conductive properties of the body [FERG11]. In implant-to-surface communication, galvanic coupling is used to send signals from an implanted device to electrodes on the skin. This scheme uses very little power and reduces the size and complexity of the implanted device.

The final type of device shown in Figure 15.1 is the general device. These are devices with processing and communications capability that can be incorporated into an IoT. A good example is smart home technology that can integrate virtually every device in the home into a network for central or remote control.

Figure 15.2 provides an overview of the elements of interest in IoT. The various ways that physical devices can be connected are shown on the left side of the figure. It is assumed that one or multiple networks support communication among the devices.

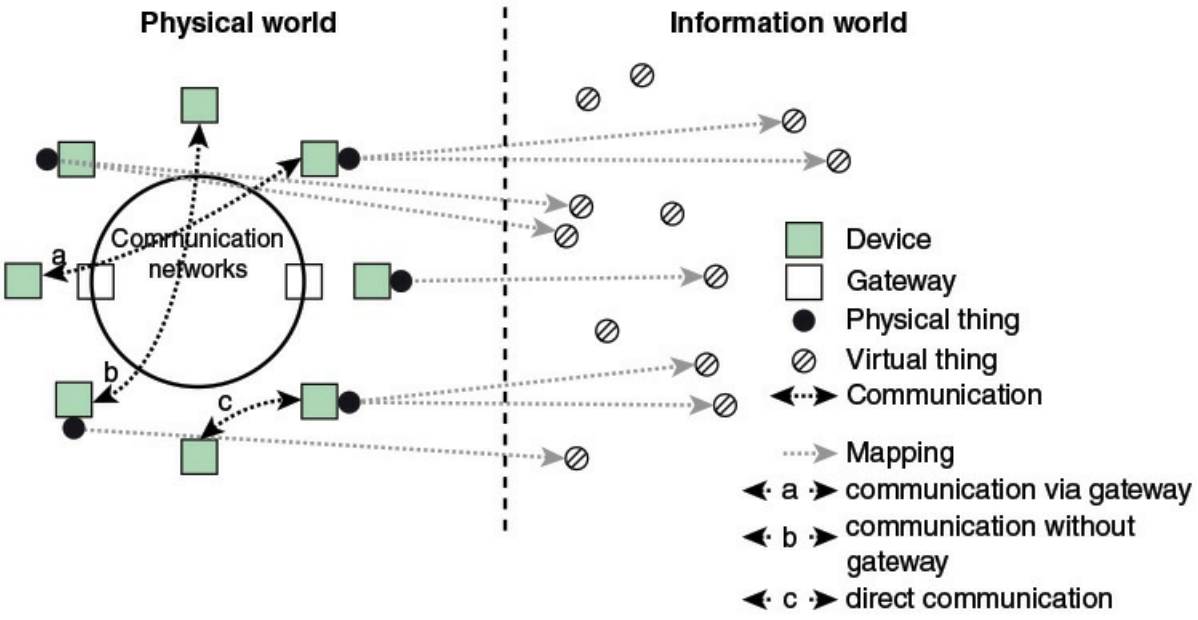

FIGURE 15.2 Technical Overview of the IoT (Y.2060)

Figure 15.2 introduces one additional IoT-related device: the gateway. At a minimum, a gateway

functions as a protocol translator. Gateways address one of the greatest challenges in designing for IoT, which is connectivity, both among devices and between devices and the Internet or enterprise network. Smart devices support a wide variety of wireless and wired transmission technologies and networking protocols. Further, these devices typically have limited processing capability. Y.2067, *Common Requirements and Capabilities of a Gateway for Internet of Things Applications*, June 2014, lays out the requirements for IoT gateways, which generally fall into three categories:

- The gateway supports a variety of device access technologies, enabling devices to communicate with each other and across an Internet or enterprise network with IoT applications. The access schemes could include, for example, ZigBee, Bluetooth, and Wi-Fi.
- The gateway supports the necessary networking technologies for both local and wide-area networking. These could include Ethernet and Wi-Fi on the premises, and cellular, Ethernet, digital subscriber line (DSL), and cable access to the Internet and wide-area enterprise networks.
- The gateway supports interaction with application, network management, and security functions.

The first two requirements involve protocol translation between different network technologies and protocol suites. The third requirement is generally referred to as an **IoT agent** function. In essence, the IoT agent provides higher-level functionality on behalf of IoT devices, such as organizing/summarizing data from multiple devices to pass on to IoT applications, implementing security protocols and functions, and interacting with network management systems.

At this point, it should be noted that the term *communication network* is not directly defined in the Y.206x series of IoT standards. The communication network or networks supports communication among devices and may directly support application platforms. This may be the extent of a small IoT, such as a home network of smart devices. More generally, the device networks connect to enterprise networks or the Internet for communication with systems that host apps and servers that host databases related to the IoT.

We can now return to the left side of Figure 15.2, which illustrates the communication possibilities among devices. The first possibility is for communication between devices via the gateway. For example, a sensor or actuator with Bluetooth capability could communicate with a data-capturing device or general device that uses Wi-Fi by means of the gateway. The second possibility is communication across the communication network without a gateway. For example, all the devices in a smart home network may use Bluetooth and could be managed from a Bluetooth-enabled computer, tablet, or smartphone. The third possibility is devices that communicate directly with each other through a separate local network and then (not shown in the figure) communicate through the communication network via a local network gateway. An example of this third possibility is the following: A number of low-power sensor devices could be deployed in an extended area, such as farmland or a factory. These could communicate with one another to pass data on toward a device connected to a gateway to the communication network.

The right side of Figure 15.2 emphasizes that each physical thing in an IoT may be represented in the information world by one or more virtual things but a virtual thing can also exist without

any associated physical thing. Physical things are mapped to virtual things stored in databases and other data structures. Applications process and deal with virtual things.

The Reference Model

Figure 15.3 depicts the ITU-T IoT reference model, which consists of four layers as well as management capabilities and security capabilities that apply across layers. We have so far been considering the device layer. In terms of communications functionality, the device layer includes, roughly, the OSI physical and data link layers. We now look at the other layers.

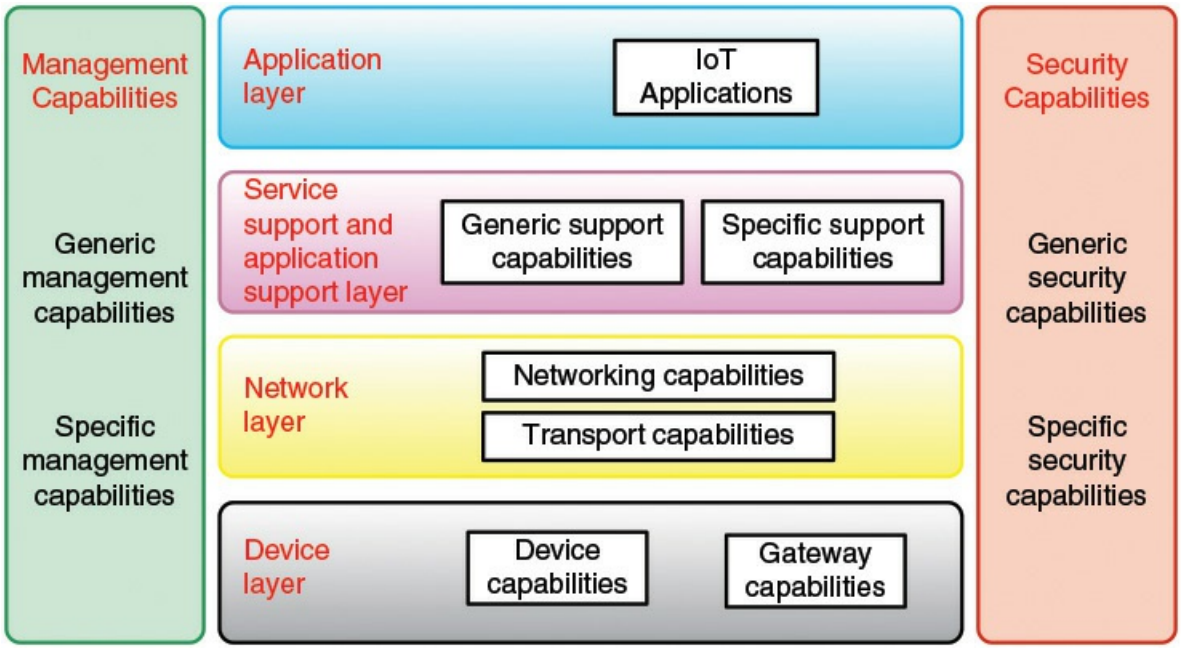

FIGURE 15.3 ITU-T Y.2060 IoT Reference Model

The **network layer** performs two basic functions. Networking capabilities refer to the interconnection of devices and gateways. Transport capabilities refer to the transport of IoT service and application specific information as well as IoT-related control and management information. Roughly, these correspond to OSI network and transport layers.

The **service support and application support layer** provides capabilities that are used by applications. Generic support capabilities can be used by many different applications. Examples include common data processing and database management capabilities. Specific support capabilities are those that cater for the requirements of a specific subset of IoT applications.

The **application layer** consists of all the applications that interact with IoT devices.

The **management capabilities layer** covers the traditional network-oriented management functions of fault, configuration, accounting, and performance management. Y.2060 lists the following as examples of generic support capabilities:

- **Device management:** Such as device discovery, authentication, remote device activation and de-activation, configuration, diagnostics, firmware/software updating, device working status management

- **Local network topology management:** Such as network configuration management
- **Traffic and congestion management:** Such as the detection of network overflow conditions and the implementation of resource reservation for time-critical/life-critical data flows

Specific management capabilities are tailored to specific classes of applications. An example is smart grid power transmission line monitoring.

The **security capabilities layer** includes generic security capabilities that are independent of applications. Y.2060 lists the following as examples of generic security capabilities:

- **Application layer:** Authorization, authentication, application data confidentiality and integrity protection, privacy protection, security audit, and antivirus
- **Network layer:** Authorization, authentication, user data and signaling data confidentiality, and signaling integrity protection
- **Device layer:** Authentication, authorization, device integrity validation, access control, data confidentiality, and integrity protection

Specific security capabilities relate to specific application requirements, such as mobile payment security requirements.

IoT World Forum Reference Model

The IoT World Forum (IWF) is an industry-sponsored annual event that brings together representatives of business, government, and academia to promote the market adoption of IoT. The IoT World Forum Architecture Committee, made up of industry leaders including IBM, Intel, and Cisco, released an IoT reference model in October 2014. This model serves as a common framework to help the industry accelerate IoT deployments. The reference model is intended to foster collaboration and encourage the development of replicable deployment models.

IoT World Forum

This reference model is a useful complement to the ITU-T reference model. The ITU-T documents focus on the device and gateway level with only a broad depiction of the upper layers. Indeed, Y.2060 describes the application layer with a single sentence. The ITU-T Y.206x series seems most concerned with defining a framework to support development of standards for interaction with IoT devices. The IWF is concerned with the broader issue of developing the applications, middleware, and support functions for an enterprise-based IoT.

Figure 15.4 depicts the seven-level model. The white paper in the IWF model issued by Cisco [CISC14b] indicates that the model is designed to have the following characteristics:

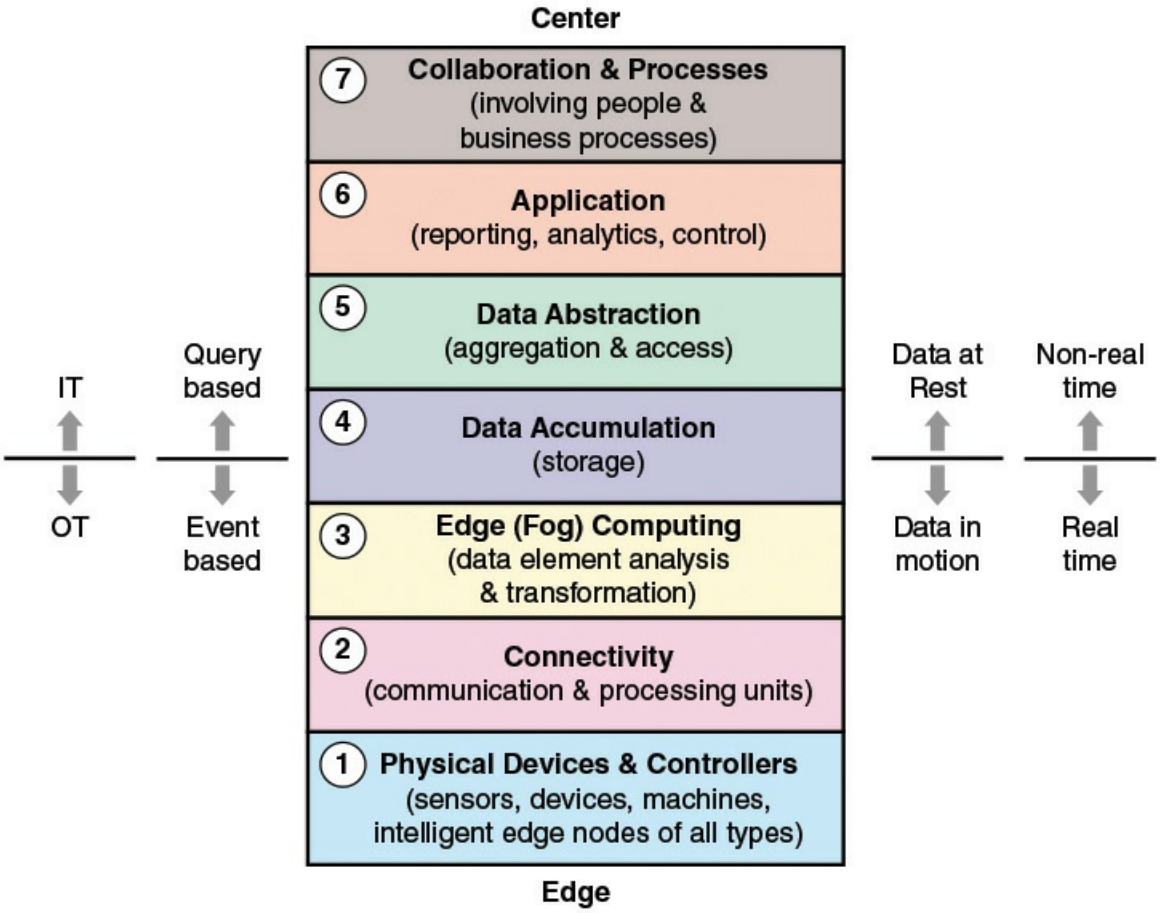

FIGURE 15.4 IoT World Forum Reference Model

- **Simplifies:** It helps break down complex systems so that each part is more understandable.
- **Clarifies:** It provides additional information to precisely identify levels of the IoT and to establish common terminology.
- **Identifies:** It identifies where specific types of processing is optimized across different parts of the system.
- **Standardizes:** It provides a first step in enabling vendors to create IoT products that work with each other.
- **Organizes:** It makes the IoT real and approachable, instead of simply conceptual.

Physical Devices and Controllers Level

Level 1 consists of physical devices and controllers that might control multiple devices. Level 1 of the IWF model corresponds approximately to the device level of the ITU-T model (Figure 15.3). As with the ITU-T model, the elements at this level are not physical things as such, but rather devices that interact with physical things, such as sensors and actuators. Among the capabilities that devices may have are analog-to-digital and digital-to-analog conversion, data generation, and the ability to be queried/controlled remotely.

Connectivity Level

From a logical point of view, this level enables communication between devices and communication between devices and the low-level processing that occurs at level 3. From a physical point of view, this level consists of networking devices, such as routers, switches, gateways, and firewalls that are used to construct local and wide-area networks and provide Internet connectivity. This level enables devices to communicate with one another and to communicate, via the upper logical levels, with application platforms such as computers, remote control devices, and smartphones.

Level 2 of the IWF model corresponds approximately to the network level of the ITU-T model. The main difference is that the IWF model includes gateways in level 2 whereas the ITU-T model puts the gateway at level 1. Because the gateway is a networking and connectivity device, its placement and level 2 seems to make more sense.

Edge Computing Level

In many IoT deployments, massive amounts of data may be generated by a distributed network of sensors. For example, offshore oil fields and refineries can generate a terabyte of data per day. An airplane can create multiple terabytes of data per hour. Rather than store all of that data permanently (or at least for a long period) in central storage accessible to IoT applications, it is often desirable to do as much data processing close to the sensors as possible. Thus, the purpose of the edge computing level is to convert network data flows into information that is suitable for storage and higher level processing. Processing elements at this level may deal with high volumes of data and perform data transformation operations, resulting in the storage of much lower volumes of data. The Cisco white paper on the IWF model [CISC14b] lists the following examples of edge computing operations:

- **Evaluation:** Evaluating data for criteria as to whether it should be processed at a higher level
- **Formatting:** Reformatting data for consistent higher-level processing
- **Expanding/decoding:** Handling cryptic data with additional context (such as the origin)
- **Distillation/reduction:** Reducing/summarizing data to minimize the impact of data and traffic on the network and higher-level processing systems
- **Assessment:** Determining whether data represents a threshold or alert; this could include redirecting data to additional destinations

Processing elements at this level correspond to general devices in the ITU-T model (Figure 15.1; Table 15.1). Generally, they are deployed physically near the edge of the IoT network; that is, near the sensors and other data-generating devices. So, some of the basic processing of large volumes of generated data is offloaded and outsourced from IoT application software located at the center.

Processing at the edge computing level is sometimes referred to as **fog computing**. Fog computing and fog services are expected to be a distinguishing characteristic of the IoT. Figure

15.5 illustrates the concept. Fog computing represents an opposite trend in modern networking from cloud computing. With cloud computing, massive, centralized storage and processing resources are made available to distributed customers over cloud networking facilities to a relatively small number of users. With fog computing, massive numbers of individual smart objects are interconnected with fog networking facilities that provide processing and storage resources close to the edge devices in an IoT. Fog computing addresses the challenges raised by the activity of thousand or millions of smart devices, including security, privacy, network capacity constraints, and latency requirements. The term *fog computing* is inspired by the fact that fog tends to hover low to the ground, whereas clouds are high in the sky.

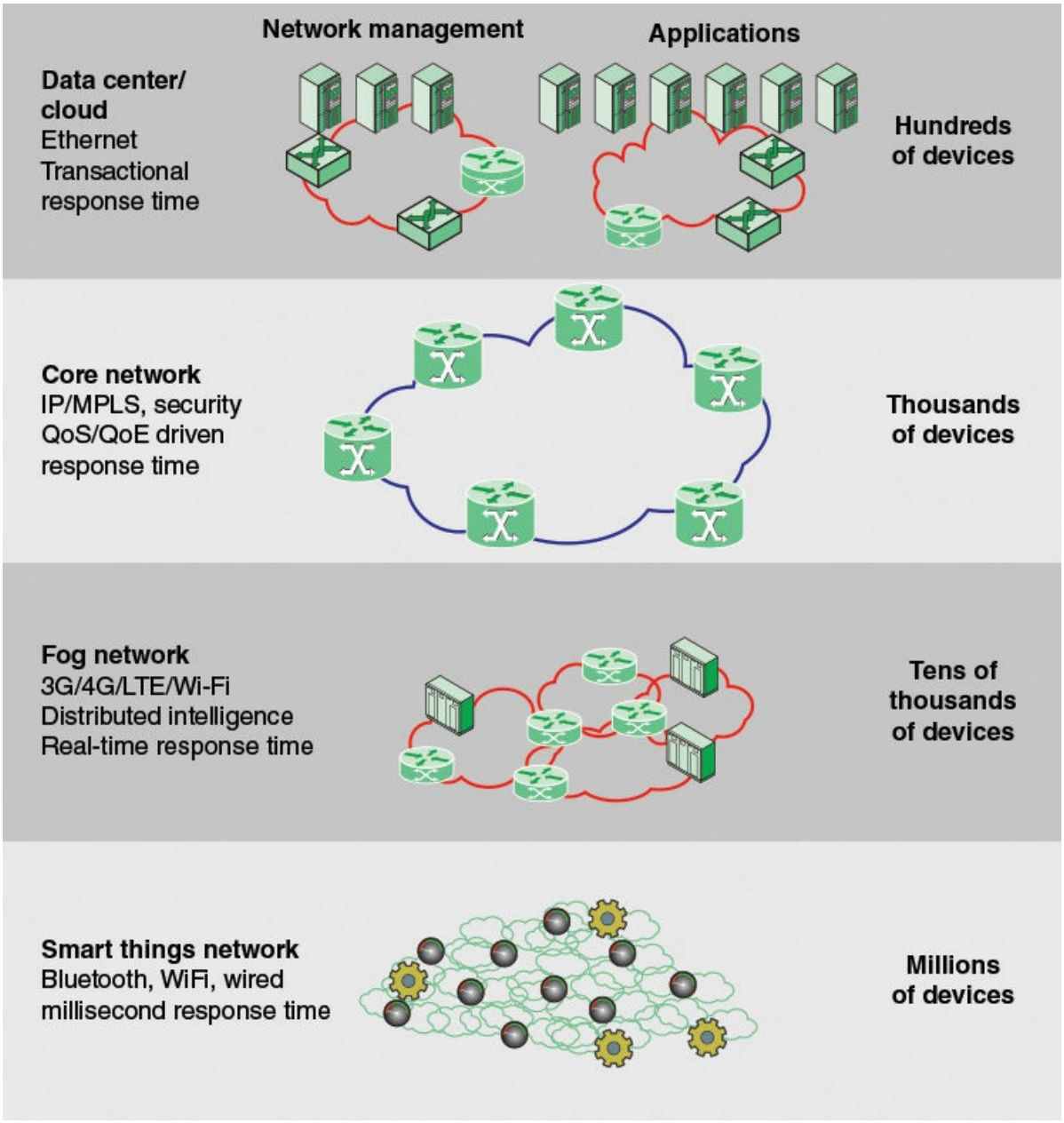

FIGURE 15.5 Fog Computing

Table 15.2, based on one in a paper by Vaquero and Rodero-Merino [VAQU14], compares cloud

and fog computing.

	Cloud	Fog
Location of processing/storage resources	Center	Edge
Latency	High	Low
Access	Fixed or wireless	Mainly wireless
Support for mobility	Not applicable	Yes
Control	Centralized/hierarchical (full control)	Distributed/hierarchical (partial control)
Service access	Through core	At the edge/on handheld device
Availability	99.99 percent	Highly volatile/highly redundant
Number of users/devices	Tens/hundreds of millions	Tens of billions
Main content generator	Human	Devices/sensors
Content generation	Central location	Anywhere
Content consumption	End device	Anywhere
Software virtual infrastructure	Central enterprise servers	User devices

TABLE 15.2 Comparison of Cloud and Fog Features

Data Accumulation Level

This level is where data coming from the numerous devices, and filtered and processed by the edge computing level, is placed in storage that will be accessible by higher levels. This level marks a clear distinction in the design issues, requirements, and method of processing between lower-level (fog) computing and upper-level (typically cloud) computing.

Data moving through a network is referred to as *data in motion*. The rate and organization of the data in motion is determined by the devices generating the data. Data generation is event driven, either periodically or by an event in the environment. To capture the data and deal with it in some fashion, it is necessary to respond in real time. By contrasts, most applications do not need to process data at network transfer speeds. As a practical matter, neither the cloud network nor the application platforms would be able to keep up with data volume generated by a huge number of IoT devices. Instead, applications deal with *data at rest*, which is data in some readily accessible storage facility. Applications can access the data as needed, on a non-real-time basis. Thus, the upper levels operate on a query or transaction basis, whereas the lower three levels operate on an event basis.

The Cisco white paper on the IWF model [CISC14b] lists the following as operations performed at the data accumulation level:

1. Converts data-in-motion to data-at-rest

2. Converts format from network packets to database relational tables
3. Achieves transition from event based to query based computing
4. Dramatically reduces data through filtering and selective storing

Another way of viewing the data accumulation level is that it marks the boundary between **IT** and **OT**.

Data Abstraction Level

The data accumulation level absorbs large quantities of data and places them in storage, with little or no tailoring to specific applications or groups of applications. A number of different types of data in varying formats and from heterogeneous processors may be coming up from the edge computing level for storage. The data abstraction level can aggregate and format this data in ways that make access by applications more manageable and efficient. Tasks involved could include the following:

1. Combining data from multiple sources. This includes reconciling multiple data formats.
2. Perform necessary conversions to provide consistent semantics of data across sources.
3. Place formatted data in appropriate database. For example, high-volume repetitive data may go into a big data system such as Hadoop. Event data would be steered to a relational database management system, which provides faster query times and an appropriate interface for this type of data.
4. Alerting higher-level applications that data is complete or had accumulated to a defined threshold.
5. Consolidating data into one place (with ETL [extract, transform, load], ELT [extract, load, transform], or data replication) or providing access to multiple data stores through data virtualization.
6. Protecting data with appropriate authentication and authorization.
7. Normalizing or denormalizing and indexing data to provide fast application access.

Application Level

This level contains any type of application that uses IoT input or controls IoT devices. Generally, the applications interact with level 5 and the data at rest, and so do not have to operate at network speeds. Provision should be available for streamlined operation that allows applications to bypass intermediate layers and interact directly with Layer 3 or even Layer 2. The IWF model does not strictly define applications, considering this beyond the scope of IWT model discussion.

Collaboration and Processes Level

This level recognizes the fact that people must be able to communicate and collaborate to make an IoT useful. This may involve multiple applications and exchange of data and control information across the Internet or an enterprise network.

Summary of the IoT Reference Model

The IWF views the IoT reference model as an industry-accepted framework aimed at standardizing the concepts and terminology associated with IoT. More importantly, the IWF model sets out the functionalities required and concerns that must be addressed before the industry can realize the value of the IoT. This model is useful both for suppliers who develop functional elements within the model and customers for developing their requirements and evaluating vendor offerings.

Figure 15.6, adapted from one in a Cisco presentation on the IWF model [CISC14c], pulls together the key concepts in the IWF model.

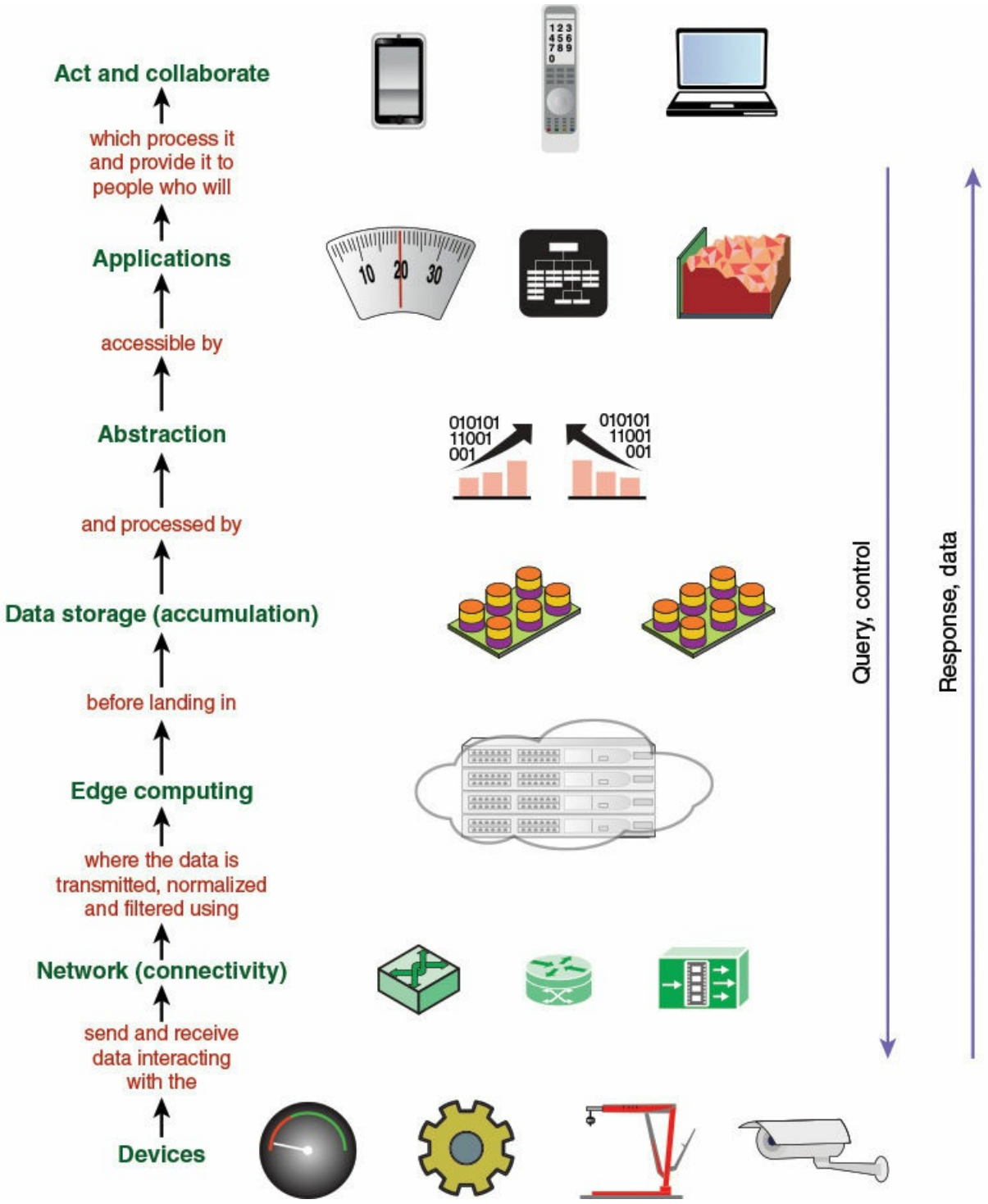

FIGURE 15.6 IoT World Forum Reference Model: Basic Premises

15.2 IoT Implementation

The preceding section looked at two reference models, which provide a good overview of the desired functionality in an IoT design. This section turns to the practical issue of deploying IoT

devices and software, by looking at three implementation efforts. First, we examine an open source software initiative, and then look at two vendor offerings.

IoTivity

IoTivity is an open source software initiative. Their objective is to provide a standard and open source implementation so devices and services will be able to work together regardless of who makes them.

IoTivity

Two organizations are playing a key role in the IoTivity project. The project is sponsored by the Open Interconnect Consortium (OIC). OIC is an industry consortium whose purpose is to promote an open source implementation to improve interoperability between the billions of devices making up the IoT. To this end, OIC is working on developing standards and an overall framework that will establish a single solution covering interoperability across multiple vertical markets and use cases. The charter of the IoTivity project is to develop and maintain an open source implementation compliant with OIC final specifications and which passes OIC certification testing.

Open Interconnect Consortium

The IoTivity Project is hosted by the Linux Foundation, the nonprofit consortium dedicated to fostering the growth of Linux and collaborative development. As a Linux Foundation project, IoTivity is overseen by an independent steering group that will work with the OIC. Developers who want to get involved with the project can access RESTful-based application programming interfaces (APIs) and submit code for peer review through the project's server. It will be made available across a range of programming languages, operating systems, and hardware platforms.

Linux Foundation

Although OIC, at the time of this writing, has not released any specifications, IoTivity has

moved forward with developing an initial “preview” release of its open source code. The initial release includes builds and Getting Started Guides for Linux, Arduino, and Tizen. The code is designed to be portable and future releases will include builds for additional operating systems.

Protocol Architecture

The IoTivity software provides a number of general-purpose query/response functions to be implemented in IoT devices and in application platforms.

IoTivity makes a distinction between a **constrained device** and an unconstrained device. Many devices in the IoT, particularly the smaller, more numerous devices, are resource constrained. As pointed out in a paper by Seghal, et al [SEGH12], technology improvements following Moore’s law continue to make embedded devices cheaper, smaller, and more energy-efficient but not necessarily more powerful. Typical embedded IoT devices are equipped with 8- or 16-bit microcontrollers that possess very little RAM and storage capacities. Resource-constrained devices are often equipped with an IEEE 802.15.4 radio, which enables low-power low-data-rate wireless personal-area networks (WPANs) with data rates of 20 to 250 kbps and frame sizes of up to 127 octets.

The term *unconstrained device* simply refers to any device without severe resources constraints. Such devices might run a general-purpose operating system, such as iOS, Android, Linux, or Windows. Unconstrained devices would include IoT devices with a good amount of processing power and memory, and application platforms for IoT applications.

To accommodate constrained devices, the overall protocol architecture (see Figure 15.7) is implemented in both constrained and unconstrained devices. At the transport level, the software relies on User Datagram Protocol (UDP), which requires minimal processing power and memory, running on top of Internet Protocol (IP). Running on top of UDP is the Constrained Application Protocol (CoAP), which is a simplified query/response protocol designed for constrained devices, and which is described subsequently. The IoTivity implementation uses libcoap, which is a C implementation of CoAP that can be used both on constrained and unconstrained devices.

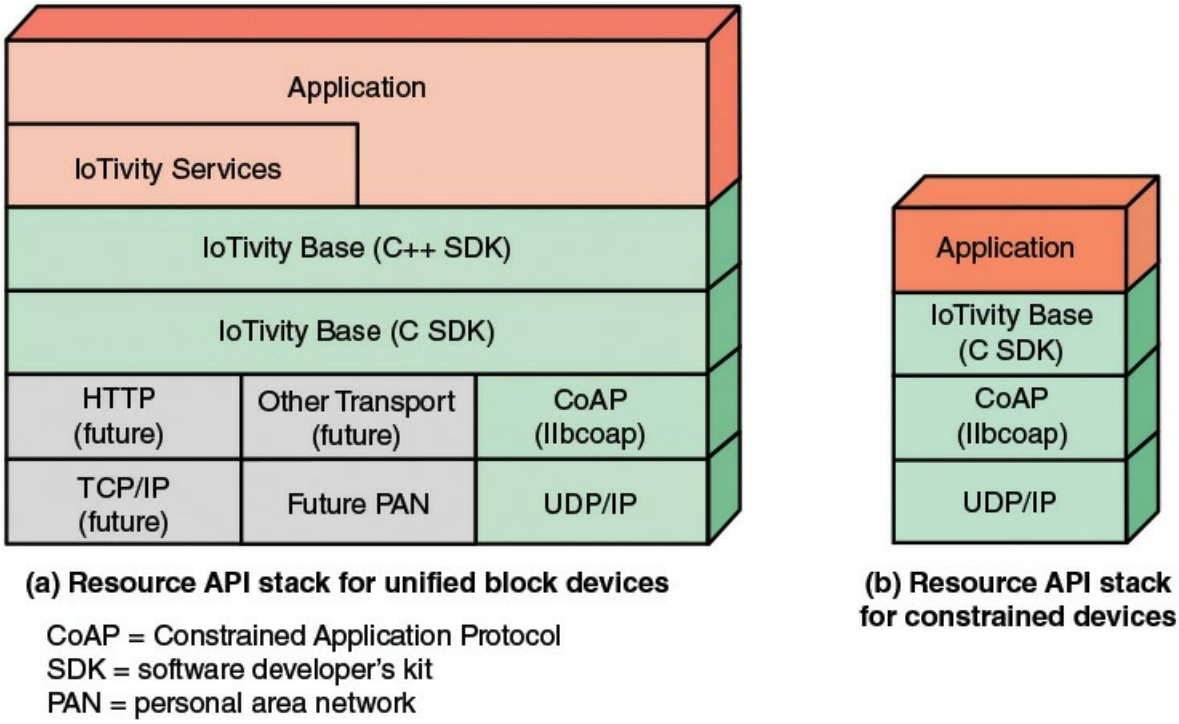

FIGURE 15.7 IoTivity Stack Blocks

The IoTivity base is a set of software development tools that support the creation of applications for communication between clients that host IoT applications and servers, which are IoT devices. The base is implemented in C, with additional tools in C++ for unconstrained devices. This software is a base for the development of open source applications that will be part of the IoTivity package, in addition to proprietary, value-added applications developed by vendors.

Constrained Application Protocol

CoAP is defined in RFC 7252, *The Constrained Application Protocol*, June 2014. The RFC describes CoAP as a specialized web transfer protocol for use with constrained nodes and constrained networks in the IoT. The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation. CoAP provides a request/response interaction model between application endpoints, supports built-in discovery of services and resources, and includes key concepts of the web such as URIs and Internet media types. CoAP is designed to easily interface with HTTP for integration with the web while meeting specialized requirements such as multicast support, very low overhead, and simplicity for constrained environments.

CoAP website

Although CoAP is designed for streamlined use in constrained devices, the protocol, with all its features, is surprisingly complex; RFC 7252 is 112 pages long. Here we provide a brief overview.

An instructive way to begin is to describe the protocol message format, shown in Figure 15.8. There are three categories of messages: Request, Response, and Empty, all of which use the same format. All messages begin with a 32-bit fixed header consisting of the following fields:

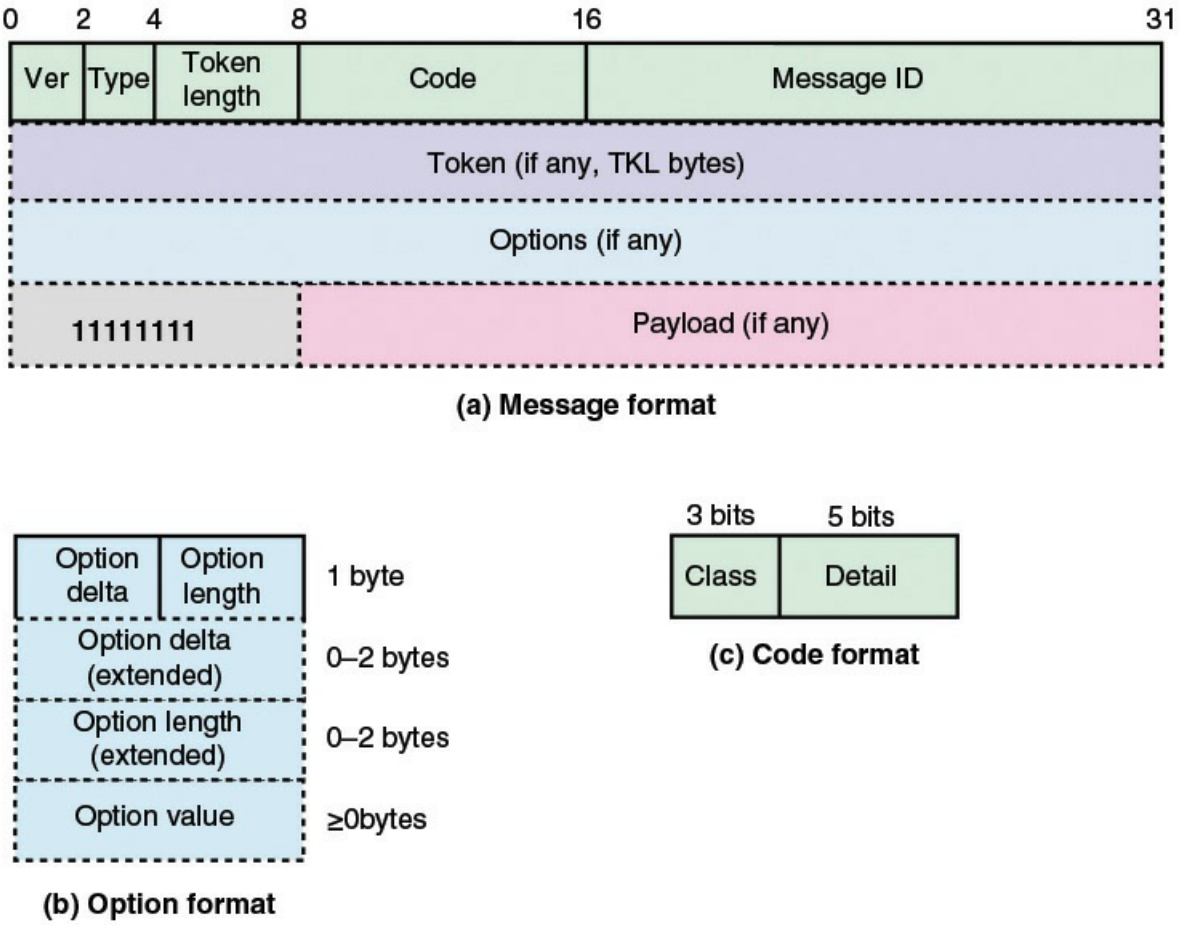

FIGURE 15.8 CoAP Formats

- **Version:** Current version is 1.
- **Type:** Message type. There are four message types:
- **Confirmable:** This message requires an acknowledgment using an ACK or Reset message. CoAP normally runs on top of UDP (UDP port number 5683), which provides an unreliable service. Thus the confirmable message type provides reliable delivery when needed.
- **Noncomfirmable:** No acknowledgment required. This is particularly true for messages that are repeated regularly for application requirements, such as repeated readings from a sensor.
- **Acknowledgment:** Acknowledges receipt of a specific confirmable message.

- **Reset:** Indicates that a specific message (confirmable or nonconfirmable) was received, but some context is missing to properly process it.
- **Token length:** Indicates the length of the variable-length token field, if any.
- **Code:** Consists of a 3-bit class and a 5-bit detail. The class indicates one of the following: request, success response, client error response, or server error response. In case of a request, the detail bits indicate the request method, which can be GET, POST, PUT, or DELETE. In case of a response, detail bits indicate the response code (see Table 15.3 and Table 15.4).

Message Class	Response Message Codes	
Request	Created	Precondition failed
Success response	Deleted	Request entity too large
Client error response	Valid	
Server error response	Changed	Unsupported content-format
Empty	Content	Internal server error
Message Types	Bad Request	Not implemented
Confirmable	Unauthorized	Bad gateway
Nonconfirmable	Bad option	Service unavailable
Acknowledgment	Forbidden	Gateway timeout
Reset	Not Found	Proxying not supported
Request Message Method Codes	Method not allowed	
GET	Not acceptable	
POST		
PUT		
DELETE		

TABLE 15.3 CoAP Messages: Classes, Types, and Codes

	Message Type			
Message Class	Confirmable	Noncomfirmable	Acknowledgment	Reset
Request	✓	✓	—	—
Success response	✓	✓	✓	—
Client error response	✓	✓	✓	—
Server error response	✓	✓	✓	—
Empty	*	—	✓	✓

— Not used

* Not used in normal operation but only to elicit a reset message (“CoAP ping”)

TABLE 15.4 CoAP Messages: Message Type Use by Message Class

- **Message ID:** Used to detect message duplication and to match messages of type acknowledgment/reset to messages of type confirmable/nonconfirmable.
- **Token:** Used to match responses to requests independently from the underlying messages. Note that the token is a concept separate from the message ID. The message ID works at the level of individual messages that require an acknowledgment. The token is intended for use as a client-local identifier for differentiating between concurrent requests (see Section 5.3); it could have been called a request ID.
- **Options:** A sequence of zero or more CoAP options in Type-Length-Value (TLV) format.

To understand the operation of CoAP, we need to distinguish among message class, message type, and message method. The message method is designed to provide a RESTful API to the next higher layer of software, and includes the typical REST functions, defined in CoAP as follows:

← *See Section 5.4, "REST"*

- **GET:** Retrieves a representation for the information that currently corresponds to the resource identified by the request URI. If the request includes an accept option, that indicates the preferred content-format of a response. If the request includes an ETag option, GET requests that ETag be validated and that the representation be transferred only if validation failed. Upon success, a content or valid response code should be present in the response message.
- **POST:** Requests that the representation enclosed in the request be processed. The actual function performed is determined by the origin server and dependent on the target resource. In essence, POST sends some data to a specified URL and, depending on context, some action is taken.
- **PUT:** Requests that the resource identified by the request URI be updated or created with the enclosed representation. In essence, PUT puts a page at a specific URL. If there's already a page there, it is replaced in its entirety. If there is no page there, a new one is created.
- **DELETE:** Requests that the resource identified by the request URI be deleted.

The simple but powerful API enables upper layer software to read and control IoT devices without worrying about the details of the protocol used to convey information. Each of the four message methods is conveyed in the Request message class and a response, if appropriate is conveyed in one of the three response message classes. Depending on the nature of the request, both the request and response may be confirmable or nonconfirmable (Table 13.8b). A response can also be carried in an acknowledgment message type (piggybacked response).

Figure 15.9, from RFC 7252, provides a simple example of CoAP message exchange. It shows a basic GET request causing a piggybacked response. The client sends a confirmable GET request for the resource coap://server/temperature to the server with a message ID of 0x7d34. The request includes one Uri-Path option (Delta 0 + 11 = 11, Length 11, Value "temperature"); the token is left empty. A 2.05 (content) response is returned in the acknowledgment message that acknowledges the confirmable request, echoing both the message ID 0x7d34 and the empty token value. The response includes a payload of "22.3 C".

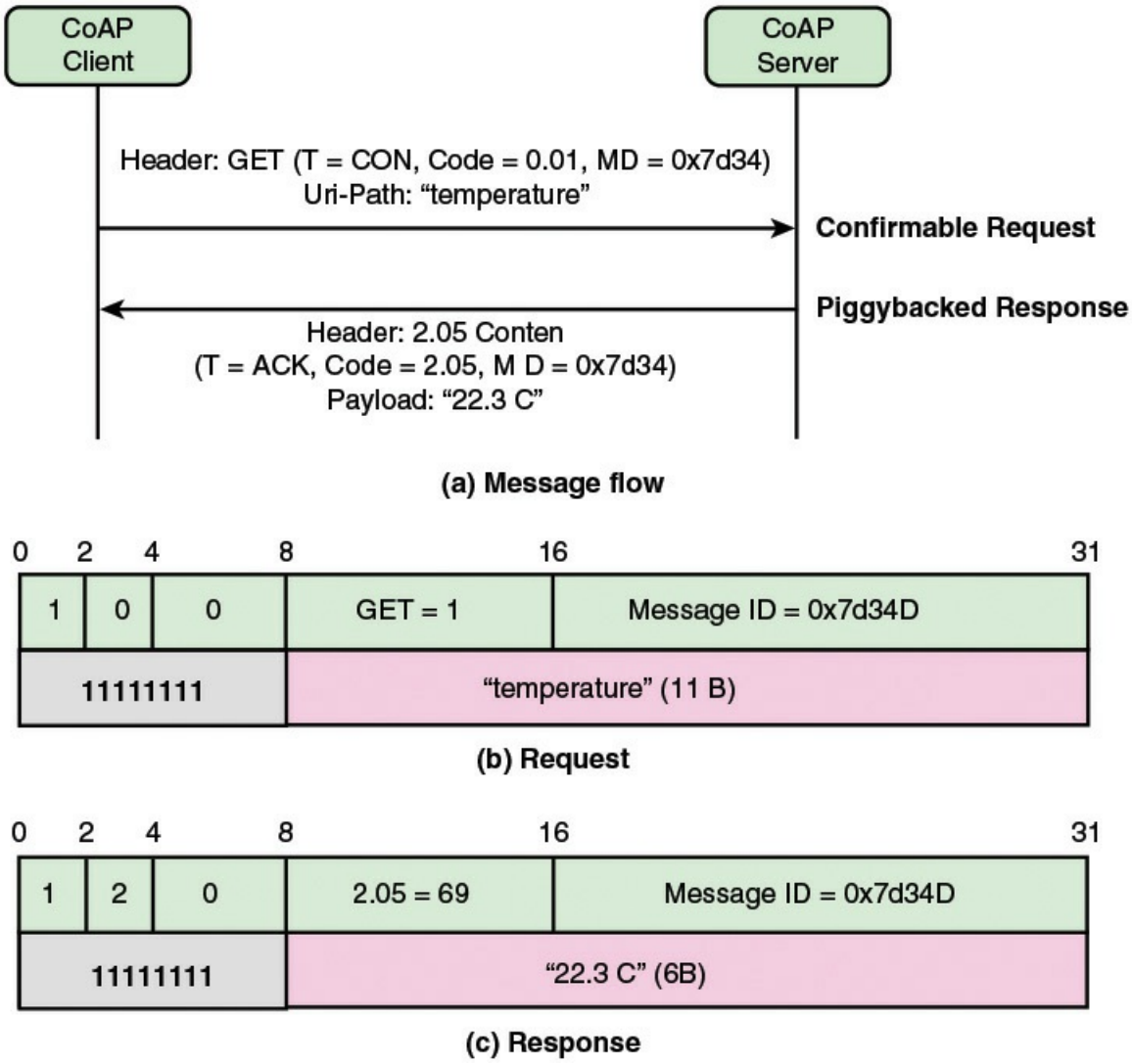

FIGURE 15.9 CoAP Example

There are other aspects of CoAP that are beyond the scope of this discussion, including security, caching, and proxy capabilities.

IoTivity Base Services

The IoTivity Base is software that runs on top of the CoAP API. It presents a resource model to higher layers, consisting of clients and servers. A server hosts resources, which are of two kinds: entity and entity handler. An entity corresponds to an IoT thing, either an actuator or a sensor. An entity handler is an associated device, such as one that caches data from one or more sensors, or a proxy for gateway type protocol conversion. The IoTivity Base provides the following services to higher layers:

- **Resource registration:** This is used to register a resource for future access.
- **Resource and device discovery:** This operation returns identification information for all

resources of a given type on the network service. The operation is sent via multicast to all services.

- **Querying resource (GET):** Get information from resource.
- **Setting a resource state (PUT):** This operation sets the value of a simple resource.
- **Observing resource state:** This operation fetches and registers as an observer for the value of a simple resource. Notifications are then provided to the client on an application-specific schedule.

The following example of querying a resource is from the IoTivity website. This example fetches the state from a light source in the following steps (see Figure 15.10):

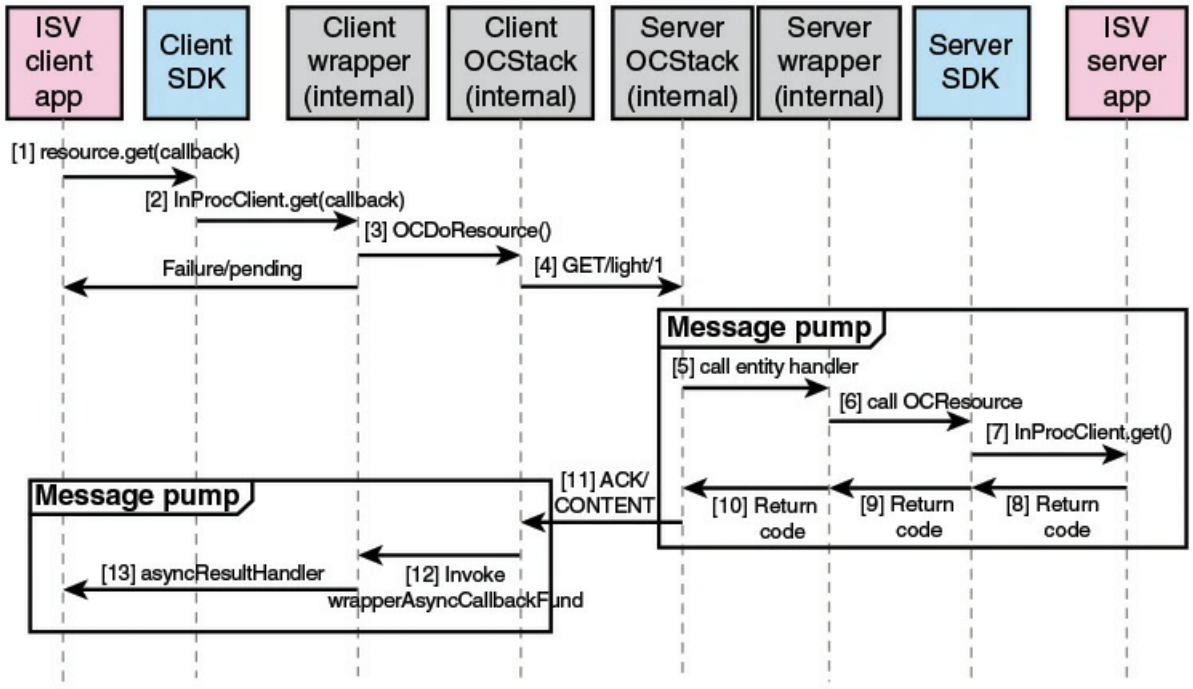

FIGURE 15.10 Sequence Diagram for Querying Resource State

1. The client application calls resource.get(...) to retrieve a representation from the resources.
2. The call is marshaled to the stack, which is either running in process or out of process (daemon).
3. The C API is called to dispatch the request. The call may look like the following: OCDoResource(OC_REST_GET, "//192.168.1.11/light/1, 0, 0, OC_CONFIRMABLE, callback);
4. Where CoAP is used as a transport, the lower stack will send a GET request to the target server.
5. On the server side, the OCProcess() function (message pump) receives and parses the request from the socket, then dispatches it to the correct entity handler based on the URI of the request.

6. Where the C++ API is used, the C++ entity handler parses the payload and marshals it to the client application depending on if the server stack is running in process or out of process (daemon).
7. The C++ SDK passes it up the C++ handler associated with the OCResource.
8. The handler returns the result code and representation to the SDK.
9. The SDK marshals the result code and representation to the C++ entity handler.
10. The entity handler returns the result code and representation to the CoAP protocol.
11. The CoAP protocol transports the results to the client device.
12. The results are returned the OCDoResource callback.
13. The results are returned to the C++ client application's syncResultCallback.

IoTivity Services

The IoTivity Base services provide a RESTful API for the basic functions outlined in the preceding subsection. On top of this, the current release includes four applications referred to as IoTivity Services. IoTivity Services provide a common set of functionalities to application development. These primitive services are designed to provide easy, scalable access to applications and resources and are fully managed by themselves. The four services are as follows:

- **Protocol Plugin Manager:** Makes IoTivity applications communicate with non-IoTivity devices by plug-in protocol converters. It provides several reference protocol plug-ins and plug-in manager APIs to start/stop plug-ins.
- **Soft Sensor Manager:** Provides physical and virtual sensor data on IoTivity in a robust manner useful for application developers. It also provides a deployment and execution environment on IoTivity for higher level virtual sensors. Its functions include the following**:** collect physical sensor data; manipulate collected data by aggregating based on its own composition algorithms; providing data to applications; detect specific events and changes.
- **Things Manager:** Creates groups, finds appropriate member things in the network, manages member presence, and makes group action easy. This service eases the task of applications by enabling them to deal with a group of things with single commands/responses.
- **Control Manager:** provides framework and services to implement a controller, a controllee, and REST framework for a controller. It also provides APIs for application developers.

To provide a better understanding of IoTivity, let's consider one of these services, the Control Manager (CM), shown in Figure 15.11. The CM runs on top of the IoTivity Base on both client and server platforms. CM provides software developer kit (SDK) APIs for discovery of controlled devices and controlling them with RESTful resource operations. CM also provides subscription/notification functionality for monitoring the device operations or state changes.

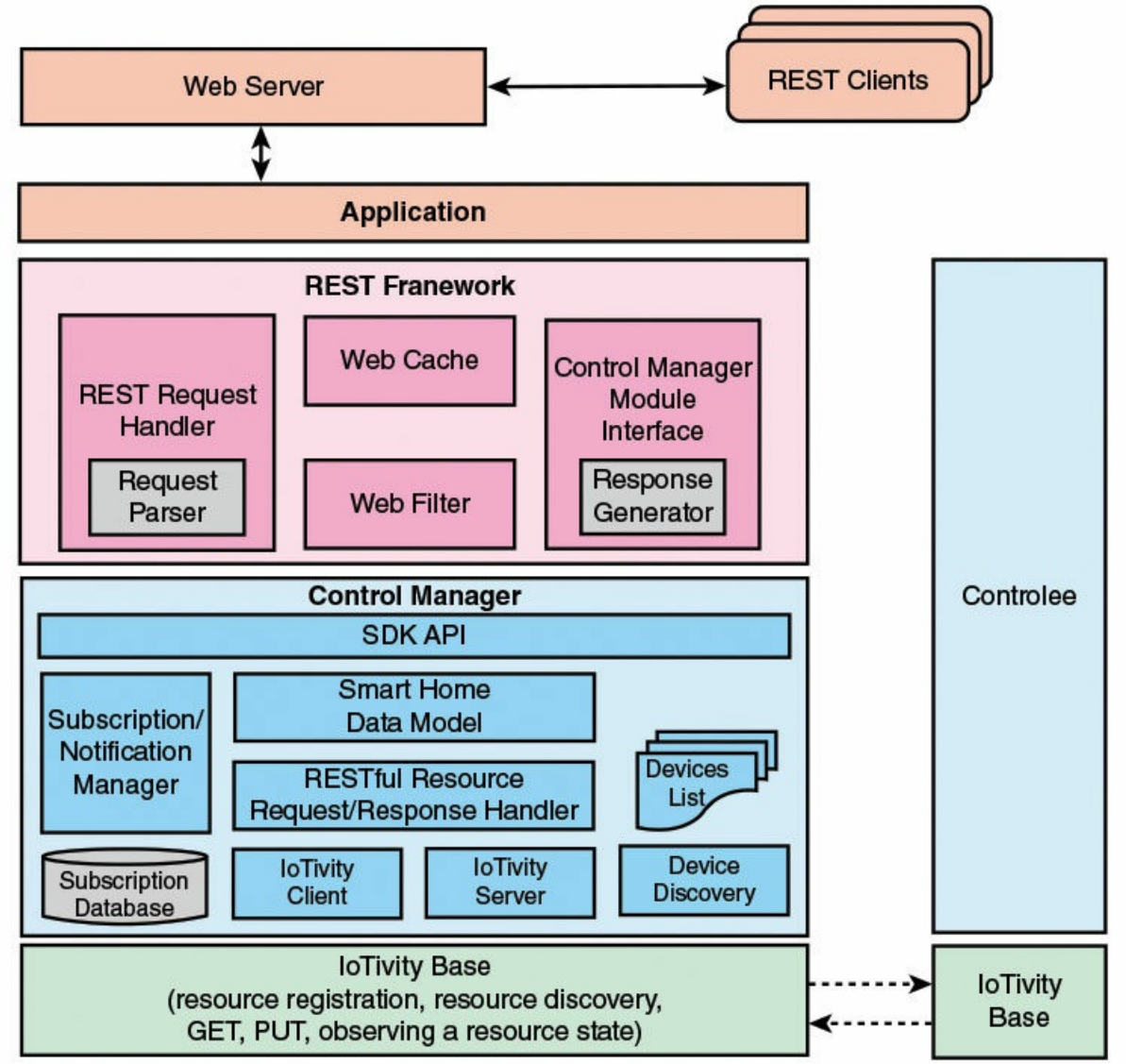

FIGURE 15.11 IoTivity Control Manager Architecture

In its current release, the CM is best suited to a smart home application. The CM makes use of the Samsung Smart Home Profile. Samsung introduced the Samsung Smart Home in early 2014. It is a service enabling Smart TVs, home appliances and smartphones to be connected and managed through a single integrated platform. Its functionality enables users to control and manage their home devices through a single application by connecting personal and home devices—from refrigerators and washing machines to smart TVs, digital cameras, smartphones and even the wearable device GALAXY Gear—through an integrated platform and server. Although the Samsung Smart Home was introduced by Samsung as a platform for controlling Samsung devices, the profile that defines the functionality can be used in other contexts and was adopted by IoTivity as an effective basis for its CM application.

The CM includes the following components:

- **SDK API:** A RESTful interface for the REST framework, discussed later in this chapter.

- **Smart home data model:** A data schema for all the available home devices and appliances, defining a hierarchical resource model and device attributes. A common set of resources provides information related to device capabilities, device configuration and supported resources. Function specific resources provide resources specific to a device function such as thermostat, light, door, and so on. With the help of a data model, application developers can easily compile device information, state and control the device.
- **RESTful resource request/response handler:** Provides the functionality of sending the requests from the controller to the controlled device by serializing it from the data model to a message format. It translates received response messages to the smart home data model for delivery to the controller. It uses the Client module for sending requests and receiving responses.
- **IoTivity client:** Implements the client using the IoTivity base framework for performing messaging with other IoTivity devices per IoTivity protocol. It supports sending requests to other IoTivity devices (for example, the controlled device) and receiving responses from them.
- **IoTivity server:** Implements the server using the IoTivity base framework for responding to requests from other IoTivity devices. CM acts like a server for responding to the discovery requests from other IoTivity devices and for receiving notifications sent from other IoTivity devices.
- **Device discovery:** Uses the IoTivity discovery mechanism of the base framework for discovering other IoTivity devices. Apart from initial device discovery, the CM discovery mechanism retrieves device specific information and capability and maintains the discovered device's information in the devices list.
- **Subscription/notification manager:** Provides functionality of subscribing to other devices and receiving notifications from other devices as defined in the Samsung Smart Home Profile. This is a RESTful subscription/notification mechanism that CM subscribes to resources of other IoTivity devices. The notifying device notifies the CM server with the REST URI specified by the CM during subscription request. CM also maintains the subscription information for the devices and resources it has already subscribed.

Referring back to Figure 15.11, we see that the CM provides a set of functions specific to smart home management and builds these on top of the more primitive functions provided by the IoTivity Base. To make the CM accessible to applications using a web-style interface, the IoTivity software release includes a REST framework software layer on top of the CM. The framework includes the following modules:

- **REST Request Handler**: Receives the REST request from the Application module, parses it, validates the request body (only schema validation) and forwards the request to the CM module via its interface. REST request handler return an error in case of invalid content (invalid URI/invalid request body, and so on).
- **Web Cache**: Caches the REST requests received from application. It responds with "304 Not Modified" when there is no change in the system after the same request was processed previously.
- **Web Filter**: Parses the filter parameters from the request URI.

- **CM Module Interface**: Acts as an interface between REST framework and the CM. It is mainly responsible for forwarding the processed REST requests to the CM. It creates and registers response listeners with the CM, which uses them to respond back asynchronously. Also, a timeout of 30 seconds is maintained here, after which if no response is received from CM, an error is sent back to the application.

Figure 15.11 shows three other elements. The execution model is that clients will interact with IoTivity through a web interface to a web server using HTTP. The web server provides a user-friendly interface for enabling the user to manage smart home devices. Each user request is passed on to the Application module, parses the HTTP request to extract information (method, URI, request body, and so on) and forwards them to the REST framework REST request handler. Responses are returned via the response generator in a similar fashion.

Cisco IoT System

In 2015, Cisco introduced a suite of integrated and coordinated products known as the Cisco IoT System. The philosophy guiding product development is based on the following observations. Cisco estimates that 50 billion devices and objects will be connected to the Internet by 2020. Yet today, more than 99 percent of things in the physical world remain unconnected. To capitalize on the unprecedented opportunities presented by this wave of digitization, companies and cities are increasingly deploying IoT solutions.

However, digitization is complex. Customers are often connecting devices and objects, or converging unrelated networks, at previously unprecedented scales. Furthermore, they can only realize the value of these connections through the application of advanced data analytics, and even then, customers often still need to create a new class of intelligent applications capable of accelerating new business models or increasing productivity. And all this has to happen without sacrificing security at any point in the system, from the device to the data center and via the cloud.

Cisco IoT System addresses the complexity of digitization with an infrastructure that is designed to manage large-scale systems of diverse endpoints and platforms, and the data deluge they create. The system consists of six critical technology pillars that, when combined together into an architecture, help reduce the complexities of digitization. Cisco also announced a number of IoT products within the six pillars and will continue to roll out new products as part of the Cisco IoT System.

Figure 15.12 illustrates the six IoT system pillars as described in the list that follows.

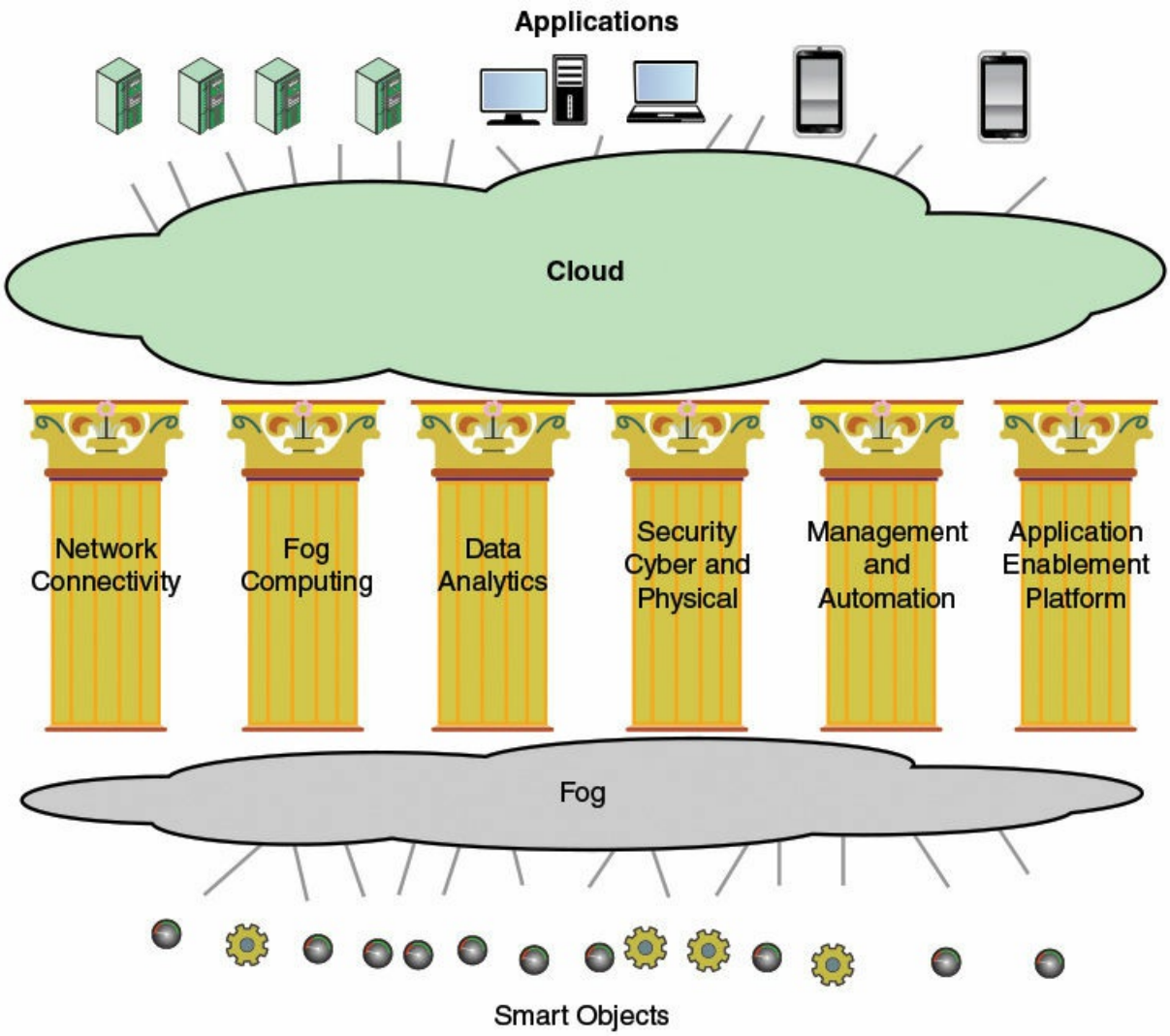

FIGURE 15.12 Cisco IoT System

- **Network connectivity:** Includes purpose-built routing, switching, and wireless products available in ruggedized and nonruggedized form factors.
- **Fog computing:** Provides Cisco's fog computing, or edge data processing platform, IOx.
- **Data analytics:** An optimized infrastructure to implement analytics and harness actionable data for both the Cisco Connected Analytics Portfolio and third-party analytics software.
- **Security:** Unifies cyber and physical security to deliver operational benefits and increase the protection of both physical and digital assets. Cisco's IP surveillance portfolio and network products with TrustSec security and cloud/cyber security products allow users to monitor, detect and respond to combined IT and operational technology (OT) attacks.
- **Management and automation:** Tools for managing endpoints and applications.
- **Application enablement platform:** A set of APIs for industries and cities, ecosystem partners and third-party vendors to design, develop, and deploy their own applications on

the foundation of IoT System capabilities.

The remainder of this discussion provides an overview of each pillar in turn. Figure 15.13, based on figures in the Cisco IoT System white paper [CISC15b], highlights key elements of each pillar.

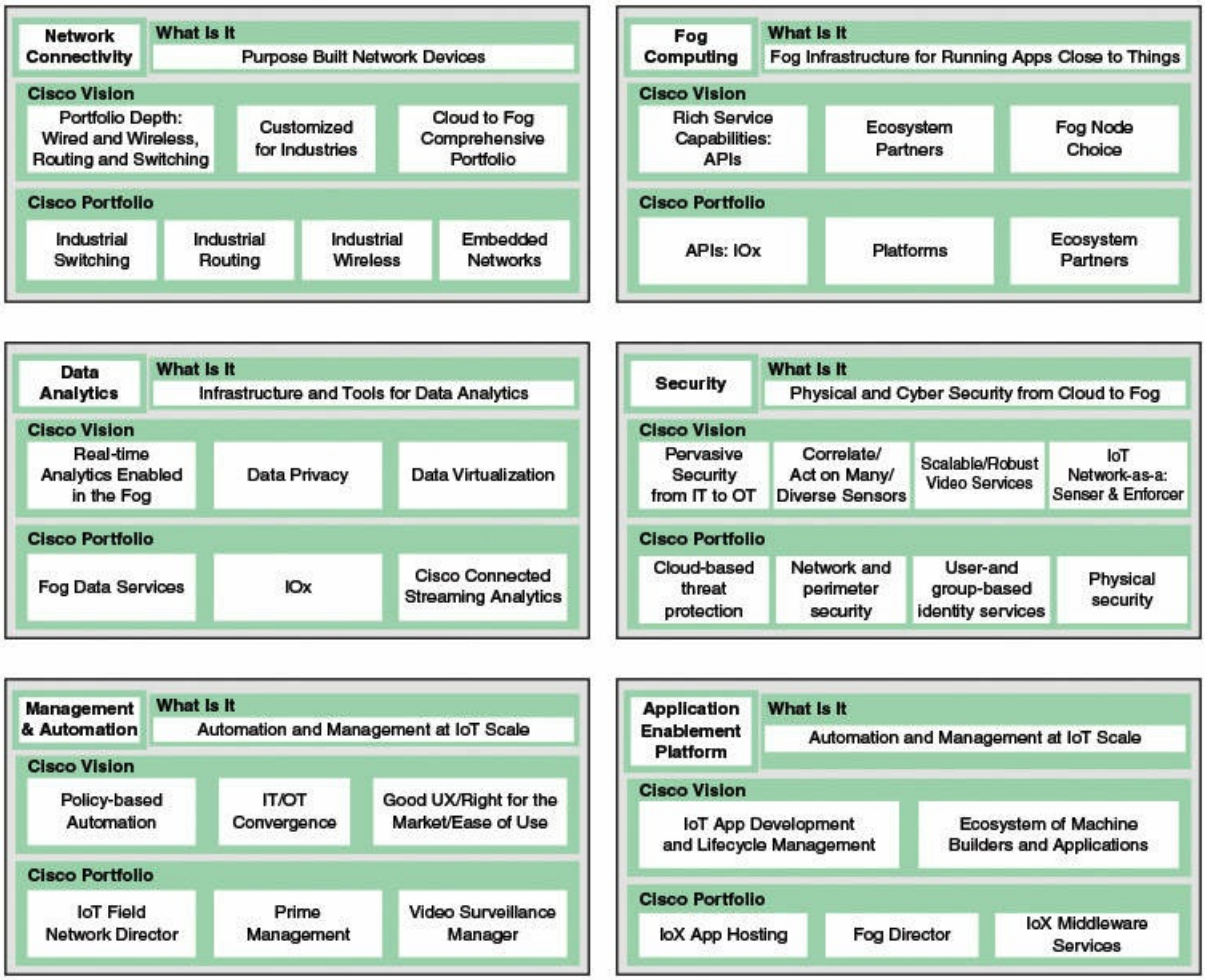

FIGURE 15.13 The Cisco IoT Pillars

Network Connectivity

The network connectivity component of Cisco IoT System is a collection of network products for the edge of the network, to support connectivity of smart objects, gateways, and other edge computing devices. Many smart objects are deployed in harsh or demanding environments, such as factories, farms, and other outdoor environments. Typically, these devices communicate wirelessly with limited transmit/receive range. Therefore, edge networking devices need to meet a number of unique requirements, including the following:

- Supporting large numbers of end systems
- Operating in demanding and possibly remote environments
- Close proximity to supported IoT objects

The network connectivity component brings together a number of preexisting and new products designed to support IoT. The product line include reliable, scalable, high-performance networking solutions with a broad portfolio of routing, switching, and wireless products, available in ruggedized and nonruggedized form factors, as well as software only solutions that integrate into third-party devices.

The product portfolio is organized into the following product categories:

- **Industrial switching:** A range of compact, ruggedized Ethernet switches that handle security, voice, and video traffic across industrial networks. A key feature of these products is that they implement Cisco's proprietary Resilient Ethernet Protocol (REP). REP provides an alternative to the Spanning Tree Protocol (STP). REP provides a way to control network loops, handle link failures, and improve convergence time. It controls a group of ports connected in a segment, ensures that the segment does not create any bridging loops, and responds to link failures within the segment. REP provides a basis for constructing complex networks and supports VLAN load balancing.
- **Industrial routing:** These products are certified to meet harsh environmental standards. They support a variety of communications interfaces, such as Ethernet, serial, cellular, WiMAX, and RF mesh.
- **Industrial wireless:** Designed for deployment in a variety of harsh or demanding environments. These products provide wireless access point functionality and implement Cisco VideoStream, which uses multicast encapsulated in unicast to improve multimedia applications.
- **Embedded networks:** Cisco Embedded Service switches are optimized for mobile and embedded networks that require switching capability in harsh environments. The primary product offering is the Cisco Embedded Service 2020 series switches product family of routers. These products are implemented on cards that can be incorporated in a variety of hardware devices. Also in this category, Cisco offers a software router application designed for small, low-powered Linux devices.

Fog Computing

The fog computing component of IoT System consists of software and hardware that extends IoT applications to the network edge, enabling data to be efficiently analyzed and managed where generated, thus reducing latency and bandwidth requirements.

The goal of the fog computing component is to provide a platform for IoT-related apps to be deployed in routers, gateways, and other IoT devices. To host new and existing applications on fog nodes, Cisco provides a new software platform, called IOx, and an API for deploying applications on IOx. The IOx platform combines the Cisco IOS operating system and Linux (see Figure 15.14). Currently, IOx is implemented on Cisco routers.

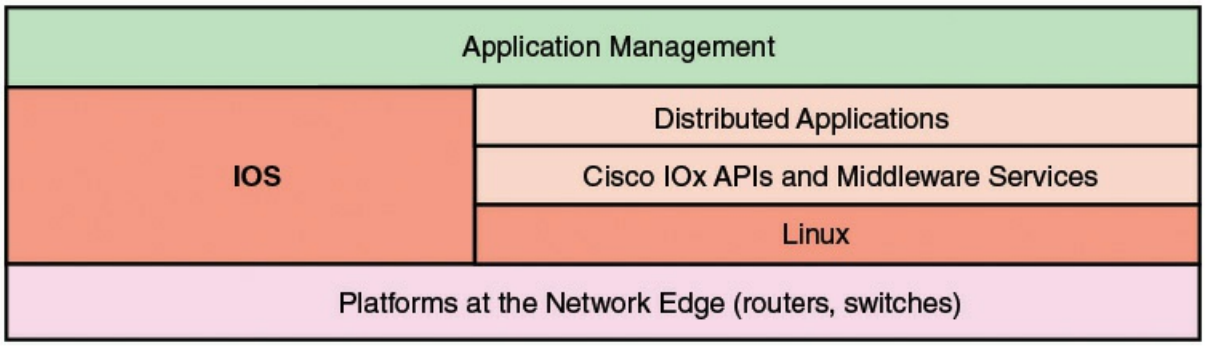

FIGURE 15.14 Cisco IOx

Cisco IOS (originally Internetwork Operating System) is software used on most Cisco Systems routers and current Cisco network switches. IOS is a package of routing, switching, internetworking, and telecommunications functions integrated into a multitasking operating system. This is not to be confused with Apple's iOS operating system that runs on iPhones and iPads.

With IOS as a base, IOx combines the communication and computing resources that are required for IoT into a single platform for application enablement at the network edge. As Figure 15.14 shows, an IOx platform, such as a router, runs IOS and Linux in parallel, using the multitasking capability of the multicore processor. Linux is used as a base to support APIs and middleware services that enable partner companies to implement fog applications on the IOx platform.

Data Analytics

The data analytics component of IoT System consists of distributed network infrastructure elements and IoT-specific APIs that run business-specific software analytics packages throughout the network architecture—from the cloud to the fog—and that allow customers to feed IoT data intelligently into business analytics.

The Cisco IoT analytics infrastructure includes the following:

- **Infrastructure for real-time analytics:** The integration of network, storage, and compute capabilities on select Cisco routers, switches, Unified Communications System (UCS) servers, and IP cameras allows analytics to run directly on fog nodes for real-time collection, storage, and analysis at the network edge.
- **Cloud to fog:** Cisco Fog Data Services includes APIs to apply business rules and control which data remains in the fog for real-time analytics and which is sent to the cloud for long-term storage and historical analysis.
- **Enterprise analytics integration:** Using IOx APIs, enterprises can run analytics on fog nodes for real-time intelligence. Fog Data Services allows IoT data exporting to the cloud. Integration of IoT data can increase operational efficiency, improve product quality, and lower costs.
- **Analytics for security:** Cisco IP cameras with storage and compute capabilities support video, audio, and data analytics at the network edge so enterprises gain real-time security intelligence, including event processing and classification.

Security

The intent of the security component is to provide solutions from the cloud to the fog that address the full attack continuum—before, during, and after an attack. The component includes cloud-based threat protection, network and perimeter security, user- and group-based identity services, video analytics, and secure physical access.

The security portfolio includes the following elements:

- **Cloud-based threat protection:** Provided by Cisco's Advanced Malware Protection (AMP) package. This is a broad spectrum of products that can be deployed on a variety of Cisco and third-party platforms. AMP products use big data analytics, a telemetry model, and global threat intelligence to help enable continuous malware detection and blocking, continuous analysis, and retrospective alerting.
- **Network and perimeter security:** Products include firewall and intrusion prevention systems.
- **User-and group-based identity services:** Products include an Identity Service Engine, which is a security policy management platform that automates and enforces context-aware security access to network resources; and Cisco TrustSec technology, which uses software-defined segmentation to simplify the provisioning of network access, accelerate security operations, and consistently enforce policy anywhere in the network.
- **Physical security:** Cisco's physical security approach consists of hardware devices and software for security management. Products include video surveillance, IP camera technology, electronic access control, and incident response. Cisco physical security solutions can be integrated with other Cisco and partner technologies to provide a unified interface that delivers situational awareness and rapid, informed decisions.

Management and Automation

The management and automation component is designed to provide simplified management of large IoT networks with support for multiple siloed functions, and to enable the convergence of OT data with the IT network. It includes the following elements:

- **IoT Field Network Director:** A software platform that provides a variety of tools for managing routers, switches, and endpoint devices. These tools include fault management, configuration management, accounting management, performance management, diagnostic and troubleshooting, and a northbound API for industry-specific applications.
- **Cisco Prime Management Portfolio:** A remote management and provisioning solution that provides visibility into the home network. The package discovers detailed information about all connected devices in the home and enables remote management.
- **Cisco Video Surveillance Manager:** Provides video, analytics and IoT sensor integration for providing physical security management.

Application Enablement Platform

This component provides a platform for cloud-based app development and deployment from cloud to fog, simply and at scale. Also offers open APIs and app development environments for use by customers, partners, and third parties. It features the following elements:

- **Cisco IOx App Hosting:** With IOx capability, customers from all segments and solution providers across industries will be able to develop, manage, and run software applications directly on Cisco industrial networked devices, including hardened routers, switches, and IP video cameras.
- **Cisco Fog Director:** Allows central management of multiple applications running at the edge. This management platform gives administrators control of application settings and lifecycle, for easier access and visibility into large-scale IoT deployments.
- **Cisco IOx Middleware Services:** Middleware is the software "glue" that helps programs and databases (which may be on different platforms) work together. Its most basic function is to enable communication between different pieces of software. This element provides tools necessary for IoT and cloud apps to communicate.

ioBridge

IoBridge provides software, firmware, and web services designed to make it simple and cost-effective to Internet-enable devices and products for manufacturers, professionals and casual users. By providing all the components necessary to web-enable things, ioBridge's customers avoid the complexity and cost associated with piecing together solutions from multiple vendors. The ioBridge offering is essentially a turnkey solution for a broad range of IoT users.

ioBridge

ioBridge Platform

IoBridge provides a complete end-to-end platform that is secure, private, and scalable for everything from do-it-yourself (DIY) home projects to commercial products and professional applications. ioBridge is both a hardware and cloud services provider. The IoT platform enables the user to create the control and monitoring applications using scalable Web technologies. ioBridge features end-to-end security, real-time I/O streaming to web and mobile apps, and easy-to-install and easy-to-use products.

Figure 15.15 illustrates some of the major features of ioBridge's technology. The tight integration between the embedded devices and the cloud services enable many of the features shown in the diagram that are not possible with traditional web server technology. Note that the off-the-shelf ioBridge embedded modules also include web-programmable control or "rules and actions." This enables the ioBridge embedded module to control devices even when it is not

connected to the ioBridge cloud server.

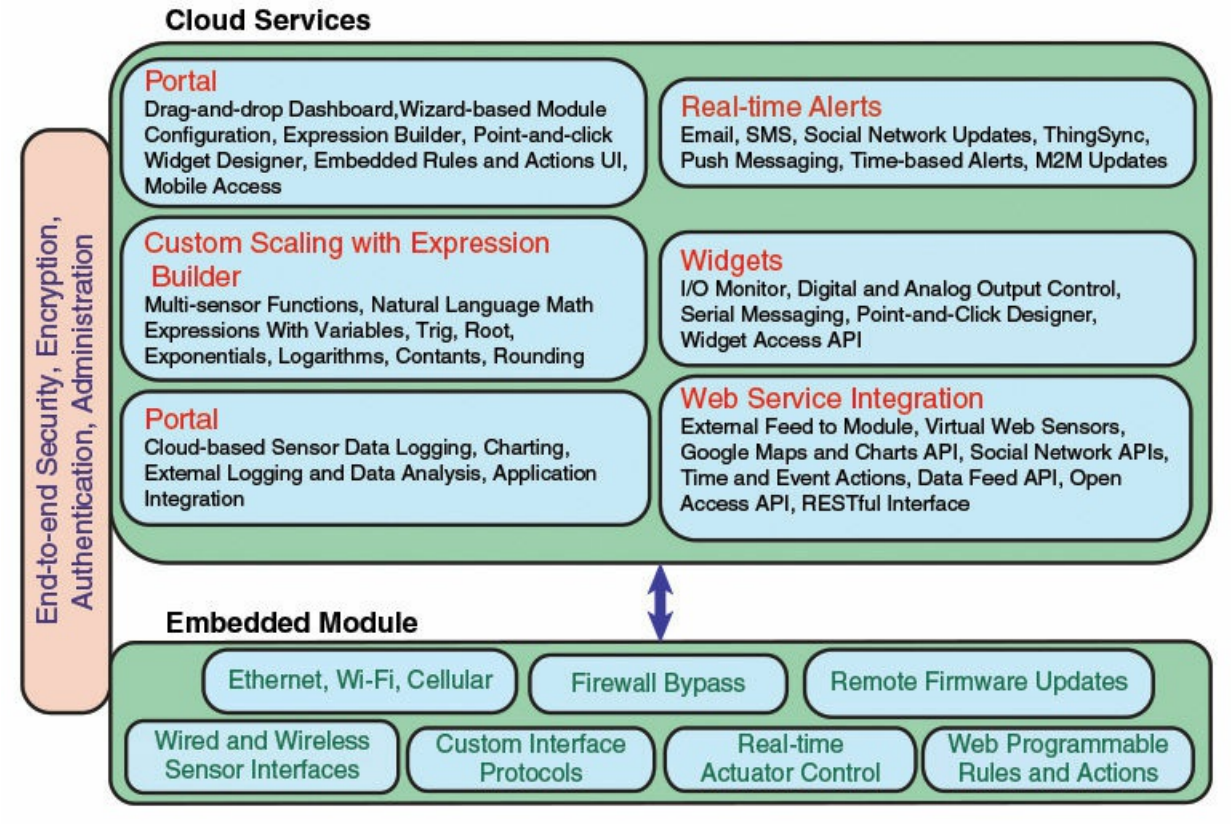

FIGURE 15.15 The ioBridge Internet of Things Platform

The major offerings on the device side are firmware, Iota modules, and gateways. Firmware is added where possible to devices to add the functionality to communicate with ioBridge services. Iotas are tiny embedded firmware or hardware modules with either Ethernet or Wi-Fi network connectivity. Gateways are small devices that can act as protocol converters and bridges between IoT devices and ioBridge services.

In essence, the IoT platform provides a seamless mashup of embedded devices with web services. IoBridge markets hardware boards, firmware, and software that can be installed in embedded devices together with apps that can run on platforms such as smartphones and tablets, as well as web services.

ThingSpeak

ThingSpeak is an open source IoT platform developed by ioBridge. ThingSpeak enables the creation of sensor logging applications, location-tracking applications, and a social network of things with status updates. It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts, the ability to create plug-ins and apps for collaborating with web services, social networks, and other APIs.

ThingSpeak

The basic element of ThingSpeak is a ThingSpeak channel, which is hosted on the ThingSpeak website. A channel stores data sent to ThingSpeak and consists of the following elements:

- **Eight fields for storing data of any type:** These can be used to store the data from a sensor or from an embedded device.
- **Three location fields:** Can be used to store the latitude, longitude and the elevation. These are very useful for tracking a moving device.
- **One status field:** A short message to describe the data stored in the channel.

IoBridge-enabled devices and platforms with ioBridge apps can communicate via a channel. A ThingSpeak channel can also connect with Twitter so that sensor updates and other data can be communicated via tweet. Note that ThingSpeak is not limited to ioBridge devices; it can work with any device that includes the software necessary to communicate via a ThingSpeak channel.

A user begins by defining a channel on the ThingSpeak website. This is an easy interactive process that includes the following steps:

1. Create new channel with unique ID.
2. Specify whether the channel will be public (open to view by anyone) or private.
3. Create from one to eight fields, which can hold any type of data, giving each field a name.
4. Create API keys. A channel has one write API key. Any data communicated to the channel will only be written into one or more fields if the data is accompanied by the API key. A channel may have multiple read API keys. If the channel is private, data can only be read by presenting the API key. A user can define an app to an API key to perform some sort of data processing or directing.

ThingSpeak provides apps that allow for an easier integration with web services, social networks, and other APIs. Some of the apps provided by ThingSpeak are the following:

- **ThingTweet:** Allows the user to post messages to twitter via ThingSpeak. In essence, this is a TwitterProxy which redirects your posts to Twitter.
- **ThingHTTP:** Allows the user to connect to web services and supports GET, PUT, POST, and DELETE methods of HTTP.
- **TweetControl:** Enables user to monitor Twitter feeds for a specific keyword and then process the request. Once the specific keyword is found in the Twitter feed, the user can then use ThingHTTP to connect to a different web service or execute a specific action.
- **React:** Sends a tweet or trigger a ThingHTTP request when the channel meets a certain condition.

- **TalkBack:** Queues up commands and then allows a device to act upon these queued commands.
- **TimeControl:** Can perform a ThingTweet, ThingHTTP, or a TalkBack at a specified time in the future. Can also be used to allow these actions to happen at a specified time throughout the week.

In addition to the listed apps, ThingSpeak allows users to create ThingSpeak applications as plug-ins using HTML, CSS, and JavaScript, which can be embedded inside a website or inside a ThingSpeak channel.

RealTime.io

Another offering of ioBridge is RealTime.io. This technology is similar to, but more powerful and sophisticated than, ThingSpeak. RealTime.io is a cloud platform that enables any device to connect to cloud services and mobile phones to provide control, alerts, data analytics, customer insights, remote maintenance, and feature selection. The intent is that product manufacturers that leverage ioBridge's technology will be able to quickly and securely bring new connected home products to market while slashing their cost-per-connected device.

RealTime.io

The RealTime.io App Builder allows the user to build web apps directly on the RealTime.io cloud platform. The user can write web applications based on HTML5, CSS, and JavaScript and create interactions with devices, social networks, external APIs, and ioBridge web services. There is an in-browser code editor, JavaScript library, app update tracking, device manager, and single sign on with existing ioBridge user accounts. RealTime.io natively works with ioBridge Iota-based devices and firmware.

RealTime.io has built-in template apps or custom apps. Template apps are prebuilt apps that the user can start with and then customize. Custom apps allow the user to upload their own files and images without any starter templates.

Figure 15.16 shows the overall ioBridge environment.

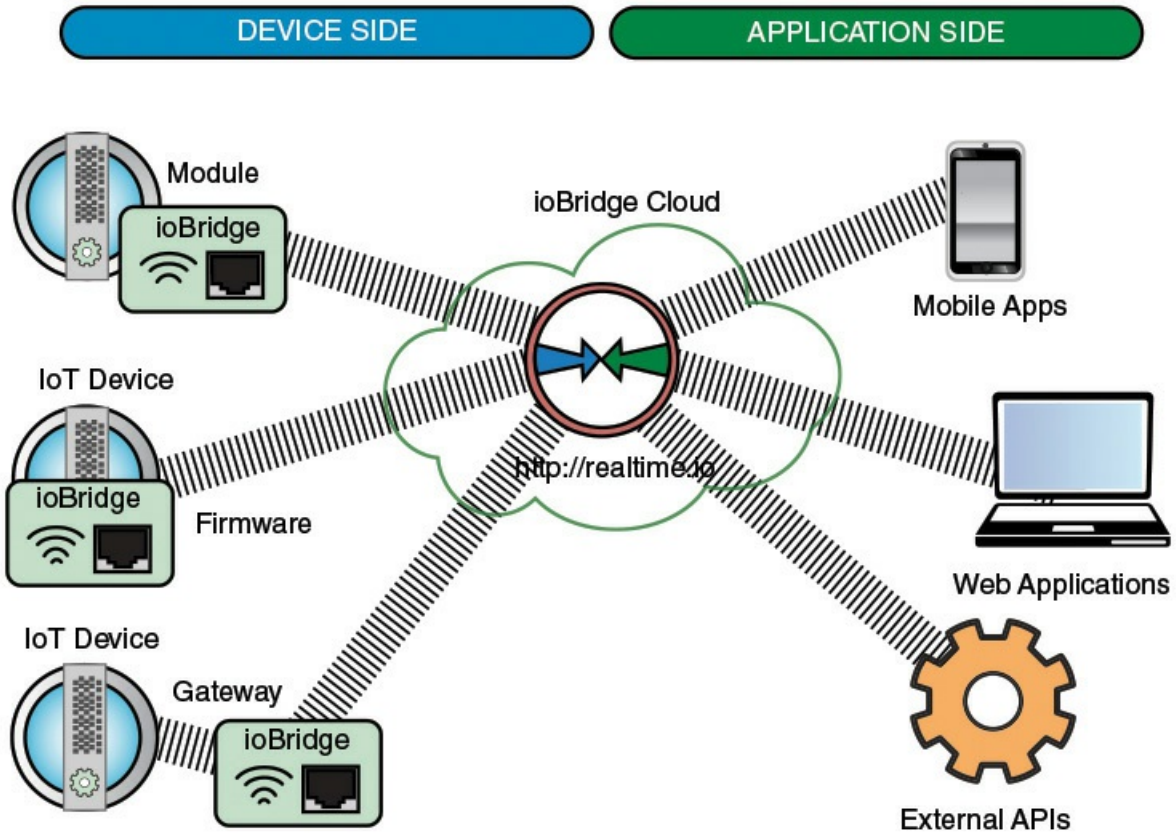

FIGURE 15.16 ioBridge Environment

15.3 Key Terms

After completing this chapter, you should be able to define the following terms.

accuracy

actuators

application processor

Constrained Application Protocol (CoAP)

constrained device

dedicated processor

deeply embedded system

electronic product code (EPC)

embedded systems

fog computing

information technology (IT)

Internet of Things (IoT)

microcontrollers

microprocessor

operational technology (OT)

precision

radio-frequency identification (RFID)

RFID reader

read range

resolution

unconstrained device

sensors

RFID tag

transceiver

15.4 References

CISC14b: Cisco Systems. *The Internet of Things Reference Model.* White paper, 2014. http://www.iotwf.com/.

CISC14c: Cisco Systems. *Building the Internet of Things.* Presentation, 2014. http://www.iotwf.com/.

CISC15b: Cisco Systems. *Cisco IoT System: Deploy, Accelerate, Innovate.* Cisco white paper, 2015.

FERG11: Ferguson, J., and Redish, A. "Wireless Communication with Implanted Medical Devices Using the Conductive Properties of the Body." *Expert Review of Medical Devices,* Vol. 6, No. 4, 2011. http://www.expert-reviews.com.

SEGH12: Seghal, A., et al. "Management of Resource Constrained Devices in the Internet of Things." *IEEE Communications Magazine,* December 2012.

VAQU14: Vaquero, L., and Rodero-Merino, L. "Finding Your Way in the Fog: Towards a Comprehensive Definition of Fog Computing." *ACM SIGCOMM Computer Communication Review,* October 2014.

Part VI: Related Topics

The reader who has persevered thus far in this account will realize the difficulties that were coped with, the hazards that were encountered, the mistakes that were made, and the work that was done.

—*The World Crisis*, Winston Churchill

Chapter 16 provides an overview of security issues that have emerged with the evolution of modern networking. Separate sections deal with software-defined networking (SDN), network functions virtualization (NFV), cloud, and IoT security, respectively. Chapter 17 concludes the book with some observations and advice about careers for the network professional.

Chapter 16. Security

> To guard against the baneful influence exerted by strangers is therefore an elementary dictate of savage prudence. Hence before strangers are allowed to enter a district, or at least before they are permitted to mingle freely with the inhabitants, certain ceremonies are often performed by the natives of the country for the purpose of disarming the strangers of their magical powers, or of disinfecting, so to speak, the tainted atmosphere by which they are supposed to be surrounded.
>
> —*The Golden Bough*, Sir James George Frazer

Chapter Objectives: **After studying this chapter, you should be able to**

- Describe the key security requirements of confidentiality, integrity, availability, authenticity, and accountability.
- Present an overview of SDN security.
- Present an overview of NFV security.
- Present an overview of cloud security.
- Present an overview of IoT security.

This chapter provides an introduction to security issues related to the main networking technologies discussed in this book. The chapter begins with a brief overview of the general security requirements that are relevant in any networking or computer environment. The remaining four sections of the chapter look at security for software-defined networking (SDN), network functions virtualization (NFV), cloud, and Internet of Things (IoT), respectively.

16.1 Security Requirements

It will be useful in the discussion in this chapter to start with an enumeration of the general security functions required to protect computer and network data and services. The five basic security functions that are widely accepted as required in most contexts consist of the following (see Figure 16.1):

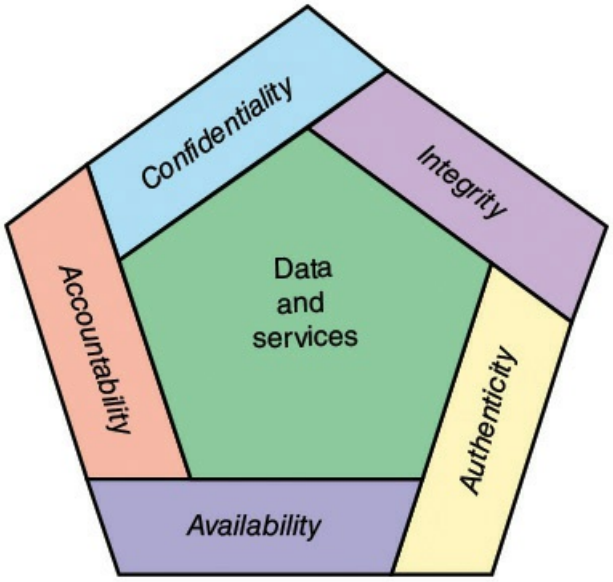

FIGURE 16.1 Essential Network and Computer Security Requirements

- **Confidentiality**: This term covers two related concepts:
  - **Data confidentiality:** Ensures that private or confidential information is not made available or disclosed to unauthorized individuals.
  - **Privacy:** Ensures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.
- **Integrity**: This term covers two related concepts:
  - **Data integrity:** Ensures that information (both stored and in transmitted packets) and programs are changed only in a specified and authorized manner.
  - **System integrity:** Ensures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.
- **Availability**: Ensures that systems work promptly and service is not denied to authorized users.
- **Authenticity**: The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator. This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source.
- **Accountability**: The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action. Because truly secure systems are not yet an achievable goal, it must be possible to trace a security breach to a responsible party. Systems must keep records of their activities to permit later forensic analysis to trace security breaches or to aid in transaction disputes.

These concepts are worth keeping in mind as we discuss the specific security requirements for

SDN, NFV, cloud, and IoT. For a more comprehensive discussion of network security, see the author's book, *Cryptography and Network Security* [STAL15b].

16.2 SDN Security

This section considers SDN security from two points of view: the security threats to SDN, and the use of SDN to enhance network security.

Threats to SDN

SDN represents a significant departure from traditional network architecture and may not mesh well with existing network security approaches. SDN involves a three-layer architecture (application, control, data) and new techniques for network control. All of this introduces the potential for new targets for attack.

Figure 16.2, from a 2014 *Network World* article [HOGG14], illustrates the potential locations of security threats in an SDN architecture. Threats can occur at any of the three layers or in the communication between layers. As shown, hardware/software platforms at any layer are potential targets for malware or intruder attacks. In addition, the protocols and application programming interfaces (APIs) related to SDN provide a new target for security attacks. This section discusses SDN-specific security threats.

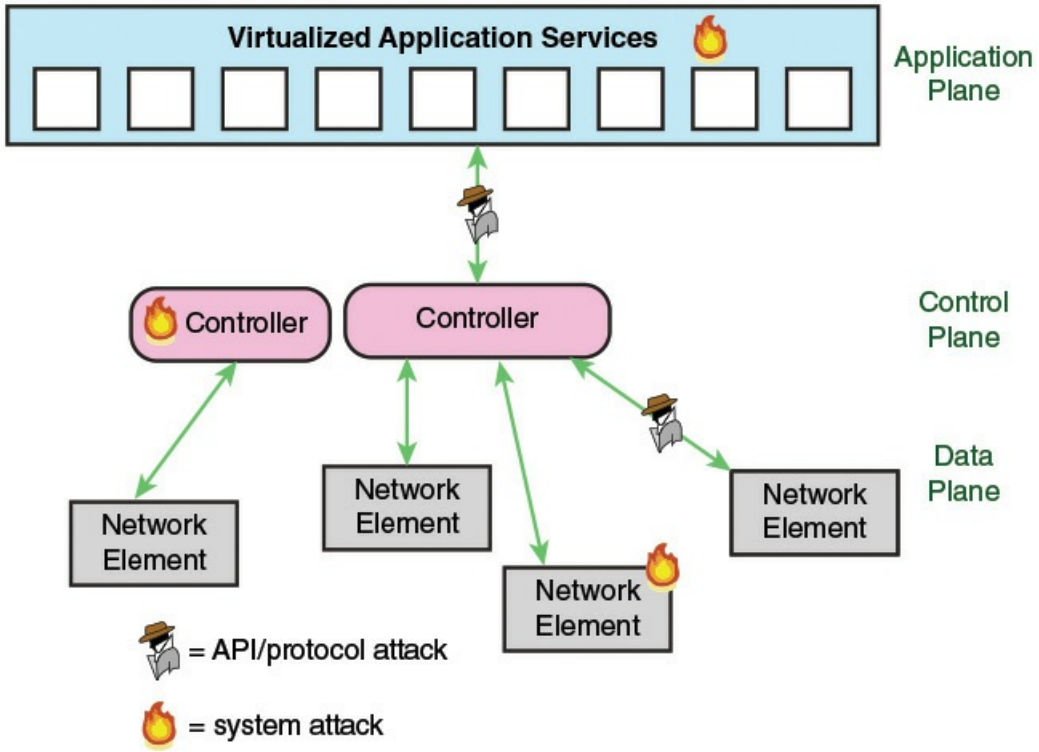

FIGURE 16.2 SDN Security Attack Surfaces

Data Plane

The key area of risk with respect to the data plane is the southbound API, such as OpenFlow and Open vSwitch Database Management Protocol (OVSDB). This API is a powerful tool for managing the data plane network elements, and increases the **attack surface** of the network infrastructure considerably because security is no longer limited to the network equipment supplier. The security of the network could be compromised by unsecure implementation of the southbound protocol. This could enable attackers to add their own flows into the flow table and spoof traffic that would otherwise be disallowed on the network. For example, the attacker might be able to define flows that bypass a firewall to introduce unwanted traffic or provide a means of eavesdropping. More generally, compromising southbound APIs would allow attackers to directly control the network elements as a whole.

One way to enhance security is the use of Transport Layer Security (TLS), which evolved from the earlier Secure Sockets Layer (SSL). Figure 14.3 illustrates the position of TLS in the TCP/IP architecture. Before discussing this architecture, we need to define the term *socket*. In essence, a socket is a method of directing data to the appropriate application in a IP-based network. The combination of the IP address of the host and a TCP or UDP port number make up a socket address. From the application point of view, a socket interface is an API. The socket interface is a generic communication programming interface implemented on UNIX and many other systems. Two applications communicate through TCP sockets. An application connects to TCP through a socket address and tells TCP what remote application is requested by means of the remote application's socket address.

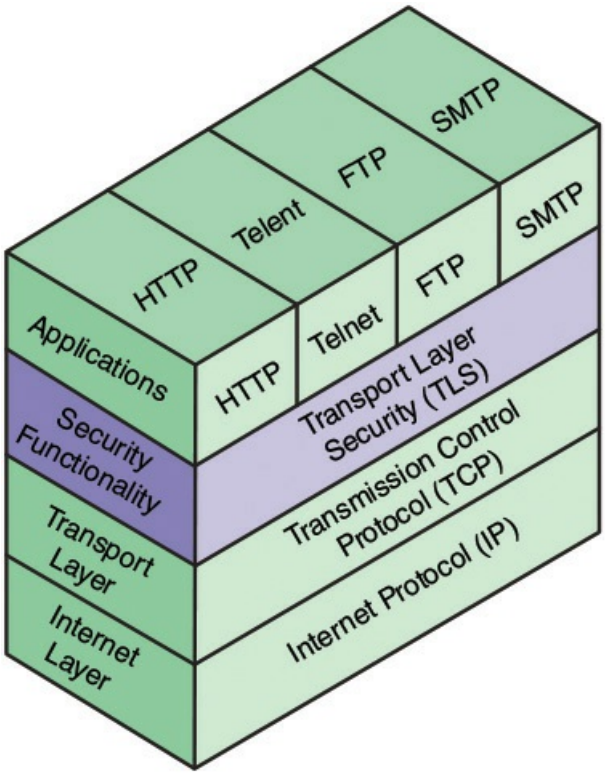

FIGURE 16.3 The Role of TLS in the TCP/IP Architecture

With TLS in place, an application has a TLS socket address and communicates to the TLS socket of the remote application. The security functions provided by TLS are transparent to the

application and also to TCP. Thus, neither TCP nor the application needs to be modified to invoke the security features of TLS. As shown in Figure 14.3, TLS supports not only HTTP but also any other application that uses TCP.

TLS provides three categories of security:

- **Confidentiality**: All data that pass between the two applications (for example, the two HTTP modules) are encrypted so that they cannot be eavesdropped.
- **Message integrity**: TLS ensures that the message is not altered or substituted for en route.
- **Authentication**: TLS can validate the identity of one or both partners to the exchange using public-key certificates. This helps prevent against a rogue controller or attacker trying to instantiate rogue flows into the network devices.

TLS consists of two phases: handshake and data transfer. During handshake, the two sides perform an authentication function and establish an encryption key to be used for data transfer. During data transfer, the two sides use the encryption key to encrypt all transmitted data.

The latest version of the OpenFlow Switch Specification, at the time of this writing (Version 1.5.1, March 26, 2015) states:

> "Between the datapath and the OpenFlow channel, the interface is implementation-specific, however all OpenFlow channel messages must be formatted according to the OpenFlow switch protocol. The OpenFlow channel is usually encrypted using TLS, but may be run directly over TCP."

However, because it is impossible to secure the data plane without securing the southbound communication channel (between control plane and data plane), TLS or an equivalent capability is necessary.

Control Plane

With SDN, the overall management, orchestration, routing, and other aspects of network traffic flow are concentrated in a single controller or a few distributed controllers. If an attacker can successfully penetrate a controller, the attacker can gain a considerable measure of control over the entire network. So, the SDN controller is a high-value target that needs a high level of protection.

Protection of the controller involves the usual repertoire of computer security techniques, including the following:

- Prevention/protection against distributed denial-of-service (DDoS) attacks. A high-availability controller architecture could go some way to mitigating a DDoS attack by using redundant controllers to make up for the loss of other controllers.
- Access control. A number of standard access control technologies can be employed, including role-based access control (RBAC) and attribute-based access control (ABAC).
- Antivirus/antiworm techniques.
- Firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS).

Application Plane

Northbound APIs and protocols present a likely target for attackers. A successful attack here could allow the attacker to gain control of the networking infrastructure. Thus, SDN security in this area focuses on preventing unauthorized users and applications from exploiting the controller. In addition, the applications themselves are a vulnerable point. If an attacker can gain control of an application and if that application is then authenticated to the control plane, the amount of damage that can be done is considerable. An authenticated application with a broad range of privileges can exercise considerable control over the configuration and operation of the network.

There are two aspects of countering these threats: Mechanisms are needed to authenticate an application's access to the control plane and prevent this authenticated application from being hacked. To counter threats throughout the authentication process involving communication between applications and the controller, the communication needs to be secured by TLS or an equivalent functionality. To protect applications, they need to be coded securely and the application platform needs to be secured against hacking.

Software-Defined Security

Although SDN presents new security challenges for network designers and managers, it also provides a platform for implementing consistent, centrally managed security policies and mechanisms for the network. SDN allows the development of SDN security controllers and SDN security applications that can provision and orchestrate security services and mechanisms.

For security management, security controllers need to provide a secure API for relevant applications. For example, as an application creates virtual machines (VMs) and configures traffic paths, it needs to be able to associate the virtual components with the appropriate security capabilities, such as IDS, IPS, and security information and event management (SIEM).

In fact, security demands may turn out to be one of the key motivating factors for deploying SDN. On the one hand, key networking trends place an increasing burden on system and networking administrators, including the following:

- The increase in network traffic volume
- The use of VMs for servers, storage, and networking devices
- Cloud computing
- The growth in the size and complexity of data centers
- The growth of IoT applications

On the other hand, there is an increasing agility and sophistication of malware. Therefore, IT manpower becomes a major security bottleneck. Security managers cannot keep up with the increasing pace of incidents and alerts and the need to fine-tune security controls in response. SDN enables security managers to bridge this response resource gap through intelligent incident detection and automated response.

The use of SDN-enabled automated tools has a benefit in itself, but this is augmented by the ability to respond on a granular basis, such as per flow, per application, or per user.

A wide variety of security applications has been developed with more on the way. A good

example is OpenDaylight's DDoS application, described in Chapter 6, "SDN Application Plane."

16.3 NFV Security

NFV dramatically changes how networks are designed, built, and managed. NFV moves network functions and network-associated functions from proprietary hardware and places them as VMs on servers that can be deployed where needed within the physical network environment. The security challenge with NFV is that it increases the attack surface and increases security complexity.

Attack Surfaces

To see this challenge, consider Figure 16.4. This repeats Figure 7.8 and indicates potential attack surfaces, as suggested in a white paper from Nakina Systems [NAKI15]. In contrast to traditional hardware-based networks, NFV blurs the hard boundaries that existed between physical network functions, making defining and administering security roles, responsibilities, and privilege levels more complex.

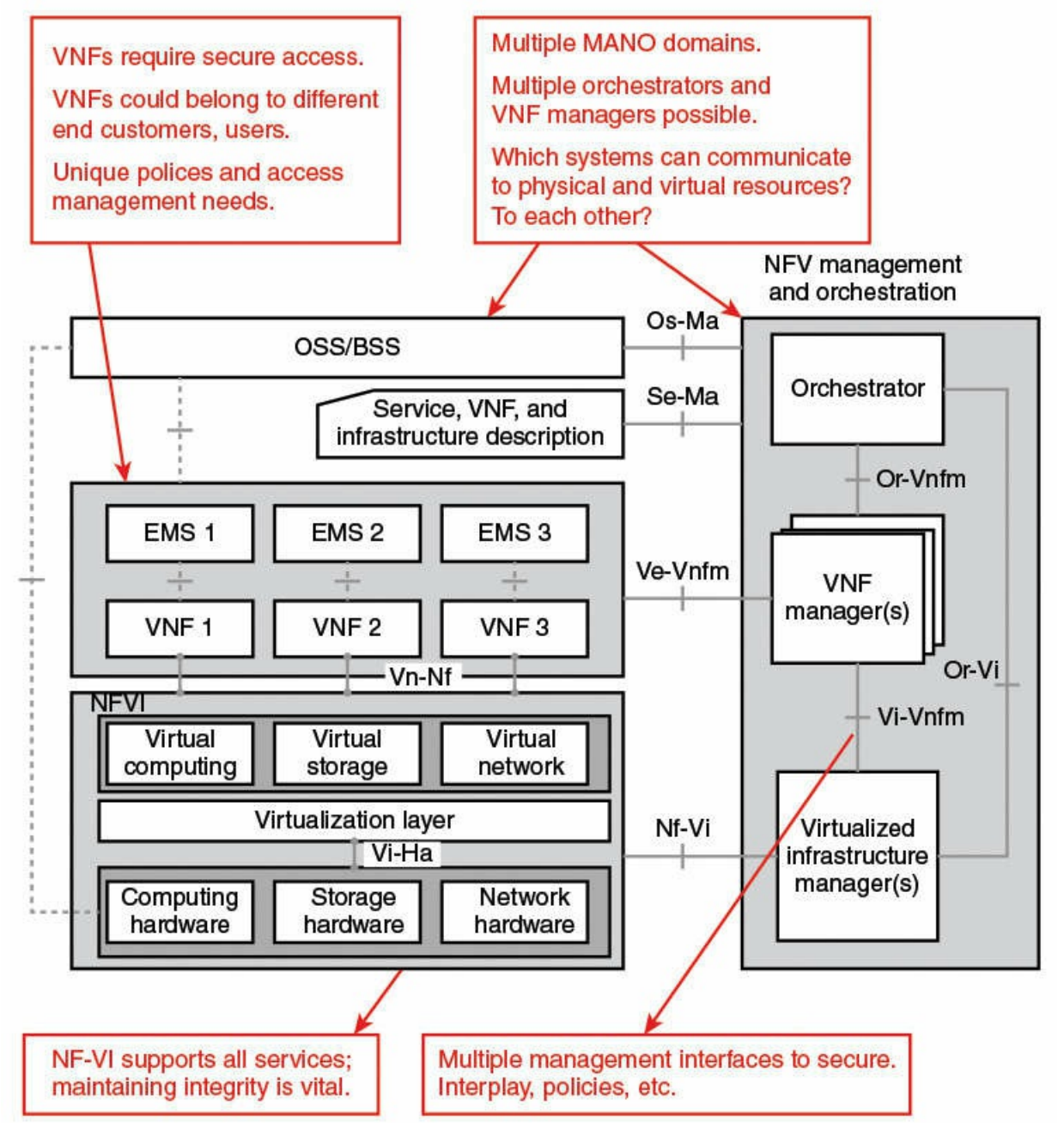

FIGURE 16.4 Potential NFV Attack Surfaces

Security needs to address multiple levels and domains and their interactions, including the following:

- **The NFV infrastructure (NFVI):** This is the domain of the underlying network, compute, and storage systems, supporting virtual computing and storage, and virtual networks.
- **Virtual network functions (VNF):** These are the network functions running on NFVI VMs.
- **MANO and OSS/BSS:** Users employ the NFV management and orchestration (MANO)

facility as well as OSS/BSS facilities to manage the network and orchestrate resources.

- **Management interfaces:** These are the critical interfaces between major domains of an NFV deployment.

A key security concern is for the system administration to control which users and/or systems can view, set, or change configuration parameters and effect network policies. This is especially important given the interdependencies between NFVIs and VNFs, and overall service performance and availability. Moreover, as multiple automated software systems access the same shared pool of network resources, ensuring that security permissions and policies do not conflict will be crucial. Software-enabled provisioning processes can lead to orchestration vulnerabilities including network configuration exploits and malicious configurations.

Figure 16.4 depicts potential NFV attack surfaces from a logical point of view. Another useful perspective is from the physical and software point of view. In particular, we are concerned with the different levels of hardware and software and what entity is in control and responsible for each element at each level. Table 16.1, which repeats Table 7.4 from Chapter 7, "Network Functions Virtualization: Concepts and Architecture," summarizes different deployment scenarios that include the physical location (building), the server hardware, the hypervisor virtualizing software, and the VNFs. Figure 16.5, which follows, illustrates these key elements.

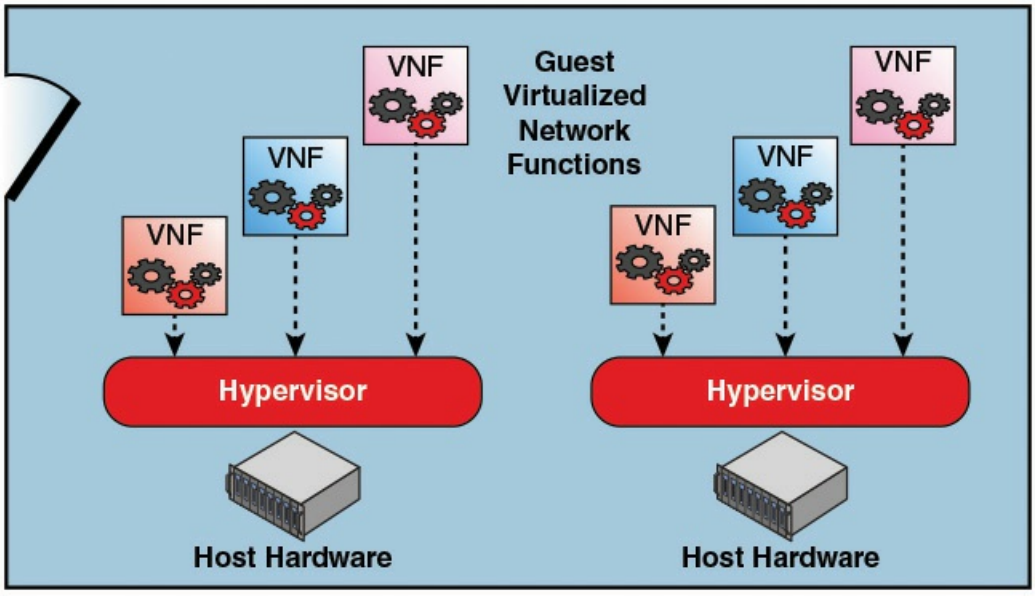

FIGURE 16.5 NFV Deployment Scenario Elements

Deployment Scenario	Building	Host Hardware	Hypervisor	Guest VNF
Monolithic operator	N	N	N	N
Network operator hosting virtual network operators	N	N	N	N, N1, N2
Hosted network operator	H	H	H	N
Hosted communications providers	H	H	H	N1, N2, N3
Hosted communications and application providers	H	H	H	N1, N2, N3, P
Managed network service on customer premises	C	N	N	N
Managed network service on customer equipment	C	C	N	N

Note: The different letters represent different companies or organizations, and are chosen to represent different roles (for example, H = hosting provider, N = network operator, P = public, C = customer). The numbered network operators (N1, N2, and so on) represent multiple individual hosted network operators.

TABLE 16.1 NFV Deployment Scenarios

Each of the levels indicated in Figure 16.5 (building, host hardware, hypervisor, VNFs) is a potential attack surface. But the design of an adequate set of security mechanisms and policies is complicated by the fact that different parties may operate at each of the levels. Therefore, the security requirements need to take this into account. Further, if there is a shared use of lower-level resources by multiple parties, then appropriate protection measures are needed. For example, if multiple VNFs from different users are running on the same physical server using the same hypervisor, then isolation of resources (for example, main memory, secondary memory, I/O ports) assigned to each user becomes a design issue.

ETSI Security Perspective

The European Telecommunications Standards Institute (ETSI), which is the lead organization in developing NFV standards, has issued four documents relating to security as part of their standards suite. The scope and field of application of each document, as defined by ETSI, is as follows:

- **NFV Security; Problem Statement (NFV-SEC 001):** Define NFV sufficiently to understand its security impact. Provide a reference list of deployment scenarios. Identify new security vulnerabilities resulting from NFV.
- **NFV Security; Cataloguing Security Features in Management Software Relevant to NFV (NFV-SEC 002):** Aims to catalogue security features in management software relevant to NFV. It covers OpenStack as the first case study. The initial deliverable is a catalogue of OpenStack modules that provide security services (such as authentication, authorization, confidentiality, integrity protection, logging, and auditing) with the full graphs of their respective dependencies down to the modules that implement cryptographic protocols and algorithms. Once the dependency graph is established, recommendations could be made on which options are appropriate for NFV deployment.

- **NFV Security; Security and Trust Guidance (NFV-SEC 003):** Define areas of consideration where security and trust technologies, practices, and processes have different requirements than non-NFV systems and operations. Supply guidance for the environment that supports and interfaces with NFV systems and operations, but avoid redefining any security considerations that are not specific to NFV.
- **NFV Security; Privacy and Regulation; Report on Lawful Interception (LI) Implications (NFV-SEC 004):** Identifies the necessary capabilities to be provided by NFV to support LI and identifies the challenges of providing LI in an NFV.

The ETSI documents classify the set of all security threats to a network comprising VNFs as illustrated in Figure 16.6 and described in the list that follows.

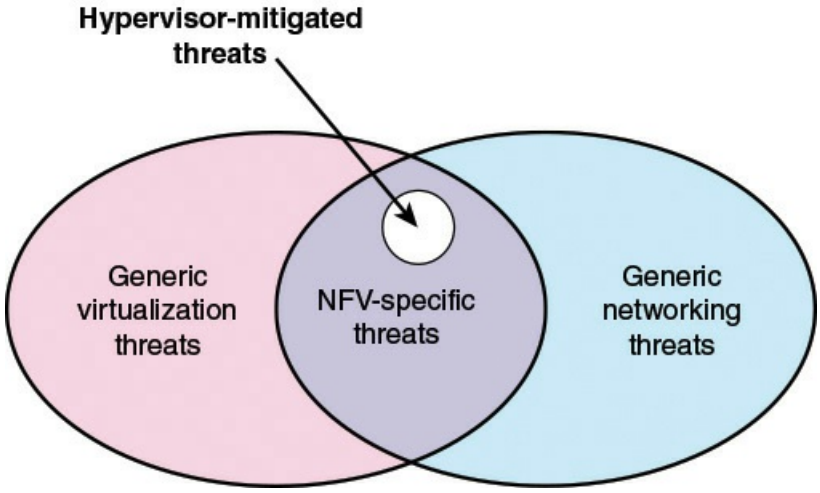

FIGURE 16.6 Classification of Threats in an NFV Networking Environment

- **Generic virtualization threats:** Threats faced by any virtualization implementation, such as failure to isolate guest users.
- **Generic networking threats:** Threats specific to the system of physical network functions prior to virtualization (for example, DDoS, firewall breach or bypass).
- **NFV-specific threats:** Threats that arise with the combining of virtualization technology and networking.

Examples of NFV-specific threats include the following:

- The use of hypervisors may introduce additional security vulnerabilities. Third-party certification of hypervisors should help shed light on their security properties. In general, to reduce the vulnerabilities of the hypervisors in use, it is essential to follow the best practices on hardening and patch management. To ensure that the right hypervisor is being executed calls for authenticating the hypervisor at the boot time through secure boot mechanisms.
- The usage of shared storage and shared networking may also add additional dimensions of vulnerability.
- The interconnectivity among NFV end-to-end architectural components (for example, hardware resources, VNFs, and management systems) exposes new interfaces that, unless

protected, can create new security threats.

- The execution of diverse VNFs over the NFV infrastructure can also create additional security issues, in particular if VNFs are not properly isolated from others.

ETSI also makes the observation that virtualization can eliminate some and mitigate other threats inherent to nonvirtualized network functions through the use of **hypervisor introspection** and other techniques. Hypervisor introspection has become a common security technique in virtualized environments. Hypervisor-based introspection can help detect attacks on VMs and guest operating systems, even when the guest operating systems are tampered with. Introspection is through monitoring of memory, program execution, access to data files, and network traffic. It can, in particular, thwart kernel-level rootkits (KLRs).

Security Techniques

It will be useful to examine a somewhat different perspective, provided in a paper by Hawilo, et al. [HAWI14]. This paper classifies the NFV environment into three functional domains, and specifies the risks and potential solutions for each as summarized in Table 16.2.

Functional Domain	Security Risks	Solutions and Requirements
Virtualized environment domain (hypervisor)	Unauthorized access or data leakage	Isolation of the served VM space, with access provided only with authentication controls.
Computing domain	Shared computing resources (CPU, memory, and so on)	Secured threads.
		Private and shared memory allocations should be erased before reallocation.
		Data should be used and stored in an encrypted manner by which exclusive access is provided only to the VNF.
Infrastructure domain	Shared logical networking layer (vSwitches)	Usage of secure networking techniques (TLS, IPsec, SSH).
	Shared physical network interface cards (NICs)	

TABLE 16.2 NFV Security Risks

16.4 Cloud Security

There are numerous aspects to cloud security and numerous approaches to providing cloud security measures. A good example of the scope of cloud security concerns and issues is seen in the NIST guidelines for cloud security, specified in SP-800-144, *Guidelines on Security and Privacy in Public Cloud Computing*, December 2011, and listed in Table 16.3. Thus, a full discussion of cloud security is well beyond the scope of this chapter. In this section, we discuss

some important cloud security topics related to the focus of this book.

Cloud Security Feature	Guidelines
Governance	Extend organizational practices pertaining to the policies, procedures, and standards used for application development and service provisioning in the cloud, as well as the design, implementation, testing, use, and monitoring of deployed or engaged services.
	Put in place audit mechanisms and tools to ensure organizational practices are followed throughout the system lifecycle.
Compliance	Understand the various types of laws and regulations that impose security and privacy obligations on the organization and potentially impact cloud computing initiatives, particularly those involving data location, privacy and security controls, records management, and electronic discovery requirements.
	Review and assess the cloud provider's offerings with respect to the organizational requirements to be met and ensure that the contract terms adequately meet the requirements.
	Ensure that the cloud provider's electronic discovery capabilities and processes do not compromise the privacy or security of data and applications.
Trust	Ensure that service arrangements have sufficient means to allow visibility into the security and privacy controls and processes employed by the cloud provider, and their performance over time.
	Establish clear, exclusive ownership rights over data.
	Institute a risk management program that is flexible enough to adapt to the constantly evolving and shifting risk landscape for the lifecycle of the system.

	Continuously monitor the security state of the information system to support ongoing risk management decisions.
Architecture	Understand the underlying technologies that the cloud provider uses to provision services, including the implications that the technical controls involved have on the security and privacy of the system, over the full system lifecycle and across all system components.
Identity and access management	Ensure that adequate safeguards are in place to secure authentication, authorization, and other identity and access management functions, and are suitable for the organization.
Software isolation	Understand virtualization and other logical isolation techniques that the cloud provider employs in its multitenant software architecture, and assess the risks involved for the organization.
Data protection	Evaluate the suitability of the cloud provider's data management solutions for the organizational data concerned and the ability to control access to data, to secure data while at rest, in transit, and in use, and to sanitize data.
	Take into consideration the risk of collating organizational data with those of other organizations whose threat profiles are high or whose data collectively represent significant concentrated value.
	Fully understand and weigh the risks involved in cryptographic key management with the facilities available in the cloud environment and the processes established by the cloud provider.
Availability	Understand the contract provisions and procedures for availability, data backup and recovery, and disaster recovery, and ensure that they meet the organization's continuity and contingency planning requirements.
	Ensure that during an intermediate or prolonged disruption or a serious disaster, critical operations can be immediately resumed, and that all operations can be eventually reinstituted in a timely and organized manner.
Incident response	Understand the contract provisions and procedures for incident response and ensure that they meet the requirements of the organization.
	Ensure that the cloud provider has a transparent response process in place and sufficient mechanisms to share information during and after an incident.
	Ensure that the organization can respond to incidents in a coordinated fashion with the cloud provider in accordance with their respective roles and responsibilities for the computing environment.

TABLE 16.3 NIST Guidelines on Cloud Security and Privacy Issues and Recommendations

The section begins with an overview of key cloud security issues and concerns. This is followed by a discussion of specific cloud security risks and their corresponding countermeasures. The next topic deals with one of the most important cloud security issues: the protection of data stored in the cloud. The discussion then introduces the concept of cloud security as a service. A final subsection looks at technical, operational, and management control functions related to cloud security.

Security Issues and Concerns

Security is important to any computing infrastructure. Companies go to great lengths to secure on-premises computing systems, so it is not surprising that security looms as a major consideration when augmenting or replacing on-premises systems with cloud services. Allaying security concerns is frequently a prerequisite for further discussions about migrating part or all of an organization's computing architecture to the cloud. Availability is another major concern: "How will we operate if we can't access the Internet? What if our customers can't access the cloud to place orders?" are common questions.

Generally speaking, such questions only arise when businesses contemplating moving core transaction processing, such as enterprise resource planning (ERP) systems, and other mission critical applications to the cloud. Companies have traditionally demonstrated less concern about migrating high maintenance applications such as e-mail and payroll to cloud service providers even though such applications hold sensitive information.

Auditability is another concern for many organizations, especially those who must comply with Sarbanes-Oxley and/or Health and Human Services Health Insurance Portability and Accountability Act (HIPAA) regulations. The auditability of their data must be ensured whether it is stored on-premises or moved to the cloud.

Before moving critical infrastructure to the cloud, businesses should perform due diligence on security threats both from outside and inside the cloud. Many of the security issues associated with protecting clouds from outside threats are similar to those that have traditionally faced centralized data centers. In the cloud, however, responsibility for assuring adequate security is frequently shared among users, vendors, and any third-party firms that users rely on for security-sensitive software or configurations. Cloud users are responsible for application-level security. Cloud vendors are responsible for physical security and some software security such as enforcing external firewall policies. Security for intermediate layers of the software stack is shared between users and vendors.

A security risk that can be overlooked by companies considering a migration to the cloud is that posed by sharing vendor resources with other cloud users. Cloud providers must guard against theft or denial-of-service attacks by their users and users need to be protected from one another. Virtualization can be a powerful mechanism for addressing these potential risks because it protects against most attempts by users to attack one another or the provider's infrastructure. However, not all resources are virtualized and not all virtualization environments are bug-free. Incorrect virtualization may allow user code access to sensitive portions of the provider's infrastructure or the resources of other users. Once again, these security issues are not unique to the cloud and are similar to those involved in managing noncloud data centers, where different

applications need to be protected from one another.

Another security concern that businesses should consider is the extent to which subscribers are protected against the provider, especially in the area of inadvertent data loss. For example, in the event of provider infrastructure improvements, what happens to hardware that is retired or replaced? It is easy to imagine a hard disk being disposed of without being properly wiped clean of subscriber data. It is also easy to imagine permissions bugs or errors that make subscriber data visible to unauthorized users. User-level encryption may be an important self-help mechanism for subscribers, but businesses should ensure that other protections are in place to avoid inadvertent data loss.

Cloud Security Risks and Countermeasures

In general terms, security controls in cloud computing are similar to the security controls in any IT environment. However, because of the operational models and technologies used to enable cloud service, cloud computing may present risks that are specific to the cloud environment. The essential concept in this regard is that the enterprise loses a substantial amount of control over resources, services, and applications but must maintain accountability for security and privacy policies.

In a 2013 report (*The Notorious Nine Cloud Computing Top Threats*), The Cloud Security Alliance [CSA13] lists the following as the top cloud-specific security threats:

- **Abuse and nefarious use of cloud computing**: For many cloud providers (CPs), it is relatively easy to register and begin using cloud services, some even offering free limited trial periods. This enables attackers to get inside the cloud to conduct various attacks, such as spamming, malicious code attacks, and denial of service. Platform as a Service (PaaS) providers have traditionally suffered most from this kind of attacks; however, recent evidence shows that hackers have begun to target Infrastructure as a Service (IaaS) vendors as well. The burden is on the CP to protect against such attacks, but cloud service clients must monitor activity with respect to their data and resources to detect any malicious behavior.

  Countermeasures include (1) stricter initial registration and validation processes, (2) enhanced credit card fraud monitoring and coordination, (3) comprehensive inspection of customer network traffic, and (4) monitoring public blacklists for one's own network blocks.

- **Unsecure interfaces and APIs**: CPs expose a set of software interfaces or APIs that customers use to manage and interact with cloud services. The security and availability of general cloud services is dependent upon the security of these basic APIs. From authentication and access control to encryption and activity monitoring, these interfaces must be designed to protect against both accidental and malicious attempts to circumvent policy.

  Countermeasures include (1) analyzing the security model of CP interfaces, (2) ensuring that strong authentication and access controls are implemented in concert with encrypted transmission, and (3) understanding the dependency chain associated with the API.

- **Malicious insiders**: Under the cloud computing paradigm, an organization relinquishes

direct control over many aspects of security and, in doing so, confers an unprecedented level of trust onto the CP. One grave concern is the risk of malicious insider activity. Cloud architectures necessitate certain roles that are extremely high-risk. Examples include CP system administrators and managed security service providers.

Countermeasures include (1) enforcing strict supply chain management and conduct a comprehensive supplier assessment, (2) specifying human resource requirements as part of legal contract, (3) requiring transparency into overall information security and management practices (as well as compliance reporting), and (4) determining security breach notification processes.

- **Shared technology issues**: IaaS vendors deliver their services in a scalable way by sharing infrastructure. Often, the underlying components that make up this infrastructure (CPU caches, GPUs, and so on) were not designed to offer strong isolation properties for a multitenant architecture. CPs typically approach this risk by the use of isolated VMs for individual clients. This approach is still vulnerable to attack, by both insiders and outsiders, and so can only be a part of an overall security strategy.

Countermeasures include (1) implementing security best practices for installation/configuration, (2) monitoring environment for unauthorized changes/activity, (3) promoting strong authentication and access control for administrative access and operations, (4) enforcing service level agreements (SLAs) for patching and vulnerability remediation, and (5) conducting vulnerability scanning and configuration audits.

- **Data loss or leakage**: For many clients, the most devastating impact from a security breach is the loss or leakage of data. We address this issue in the next section.

Countermeasures include (1) implementing strong API access control, (2) encrypting and protecting integrity of data in transit and at rest, (3) analyzing data protection at both design and run time, and (4) implementing strong key generation, storage and management, and destruction practices.

- **Account or service hijacking**: Account and service hijacking, usually with stolen credentials, remains a top threat. With stolen credentials, attackers can often access critical areas of deployed cloud computing services, allowing them to compromise the confidentiality, integrity, and availability of those services.

Countermeasures include (1) prohibiting the sharing of account credentials between users and services, (2) leveraging strong two-factor authentication techniques where possible, (3) employing proactive monitoring to detect unauthorized activity, and (4) understanding CP security policies and SLAs.

- **Unknown risk profile**: In using cloud infrastructures, the client necessarily cedes control to the cloud provider on a number of issues that may affect security. Thus the client must pay attention to and clearly define the roles and responsibilities involved for managing risks. For example, employees may deploy applications and data resources at the CP without observing the normal policies and procedures for privacy, security, and oversight.

Countermeasures include (1) disclosure of applicable logs and data, (2) partial/full disclosure of infrastructure details (for example, patch levels and firewalls), and (3) monitoring and alerting on necessary information.

Similar lists have been developed by the European Network and Information Security Agency and NIST.

Data Protection in the Cloud

There are many ways to compromise data. Deletion or alteration of records without a backup of the original content is an obvious example. Unlinking a record from a larger context may render it unrecoverable, as can storage on unreliable media. Loss of an encoding key may result in effective destruction. Finally, unauthorized parties must be prevented from gaining access to sensitive data.

The threat of data compromise increases in the cloud, due to the number of, and interactions between, risks and challenges that are either unique to the cloud or more dangerous because of the architectural or operational characteristics of the cloud environment.

Database environments used in cloud computing can vary significantly. Some providers support a **multi-instance model**, which provide a unique DBMS running on a VM instance for each cloud subscriber. This gives the subscriber complete control over role definition, user authorization, and other administrative tasks related to security. Other providers support a **multitenant model**, which provides a predefined environment for the cloud subscriber that is shared with other tenants, typically through tagging data with a subscriber identifier. Tagging gives the appearance of exclusive use of the instance, but relies on the cloud provider to establish and maintain a sound secure database environment.

Data must be secured while at rest, in transit, and in use, and access to the data must be controlled. The client can employ encryption to protect data in transit, though this involves key management responsibilities for the CP. The client can enforce access control techniques but, again, the CP is involved to some extent depending on the service model used.

For data at rest, the ideal security measure is for the client to encrypt the database and only store encrypted data in the cloud, with the CP having no access to the encryption key. So long as the key remains secure, the CP has no ability to decipher the data, although corruption and other denial-of-service attacks remain a risk.

There are a number of ways in which an encryption scheme could be implemented. A very simple arrangement is as follows. Suppose that each individual item in the database is encrypted separately, all using the same encryption key. The encrypted database is stored at the server, but the server does not have the key, so that the data are secure at the server. Even if someone were able to hack into the server's system, all he or she would have access to is encrypted data. The client system does have a copy of the encryption key. A user at the client can retrieve a record from the database with the following sequence:

1. The user issues an SQL query for fields from one or more records with a specific value of the primary key.
2. The query processor at the client encrypts the primary key, modifies the SQL query accordingly, and transmits the query to the server.
3. The server processes the query using the encrypted value of the primary key and returns the appropriate record or records.

4. The query processor decrypts the data and returns the results.

More efficient and flexible systems have been implemented. See the author's book, *Computer Security: Principles and Practice* for details [STAL15a].

Cloud Security as a Service

The term *Security as a Service* has generally meant a package of security services offered by a service provider that offloads much of the security responsibility from an enterprise to the security service provider. Among the services typically provided are authentication, anti-virus, antimalware/spyware, intrusion detection, and security event management. In the context of cloud computing, Cloud Security as a Service, designated SecaaS, is a segment of the SaaS offering of a CP.

The Cloud Security Alliance defines SecaaS as the provision of security applications and services via the cloud either to cloud-based infrastructure and software or from the cloud to the customers' on-premise systems [CSA11]. The Cloud Security Alliance has identified the following SecaaS categories of service:

- Identity and access management
- Data loss prevention
- Web security
- E-mail security
- Security assessments
- Intrusion management
- Security information and event management
- Encryption
- Business continuity and disaster recovery
- Network security

This section covers these categories with a focus on security of the cloud-based infrastructure and services (see Figure 16.7).

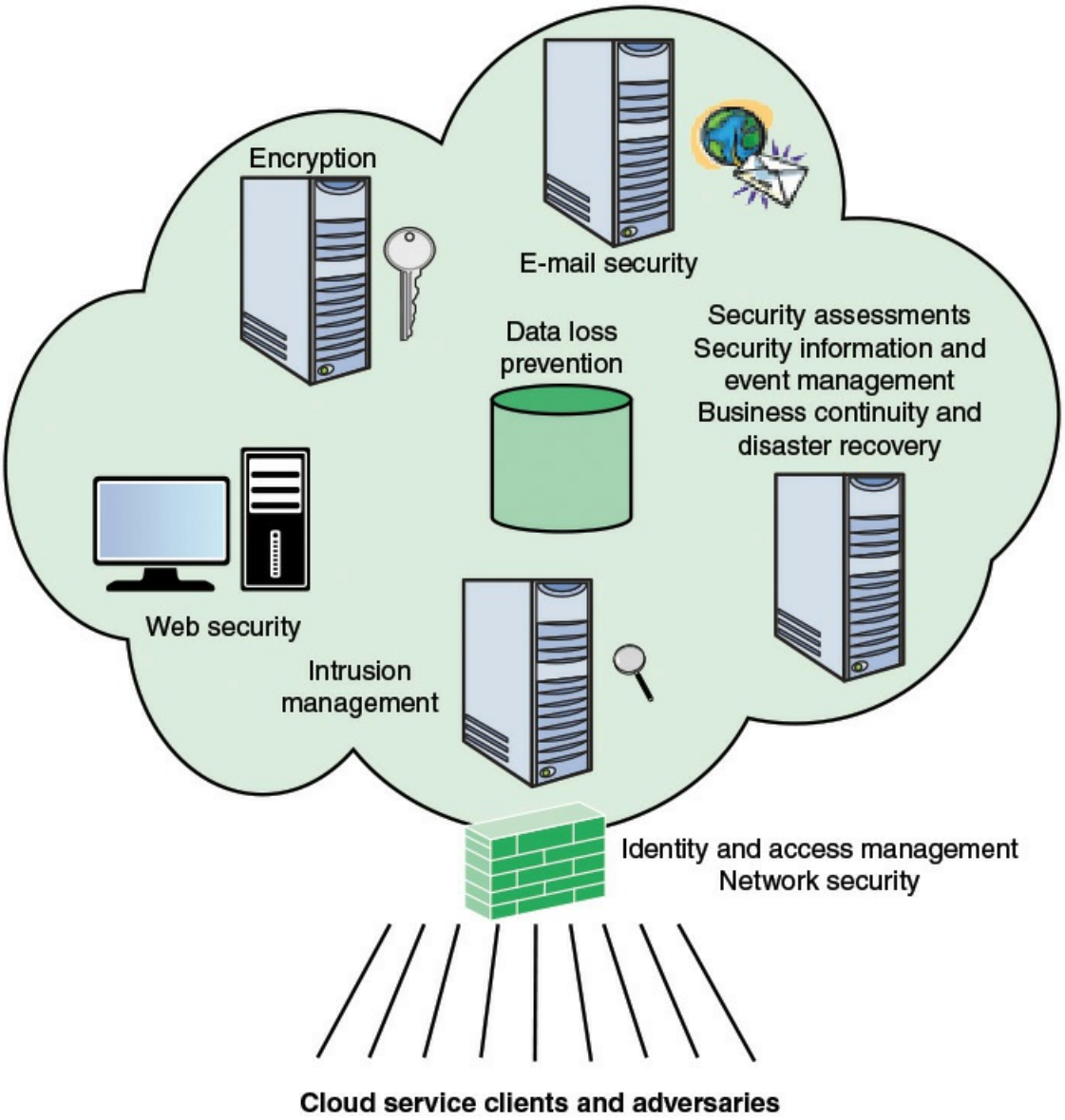

FIGURE 16.7 Elements of Cloud Security as a Service

- **Identity and access management (IAM)** includes people, processes, and systems that are used to manage access to enterprise resources by assuring that the identity of an entity is verified, and then granting the correct level of access based on this ensured identity. One aspect of identity management is identity provisioning, which has to do with providing access to identified users and subsequently deprovisioning, or denying access, to users when the client enterprise designates such users as no longer having access to enterprise resources in the cloud. Another aspect of identity management is for the cloud to participate in the federated identity management scheme used by the client enterprise. Among other requirements, the cloud service provider (CSP) must be able to exchange identity attributes with the enterprise's chosen identity provider.

The access management portion of IAM involves authentication and access control services. For example, the CSP must be able to authenticate users in a trustworthy manner.

The access control requirements in SPI environments include establishing trusted user profile and policy information, using it to control access within the cloud service, and doing this in an auditable way.

- **Data loss prevention (DLP)** is the monitoring, protecting, and verifying the security of data at rest, in motion, and in use. Much of DLP can be implemented by the cloud client, such as discussed in the preceding subsection, "Data Protection in the Cloud." The CSP can also provide DLP services, such as implementing rules about what functions can be performed on data in various contexts.
- **Web security** is real-time protection offered either on premise through software/appliance installation or via the cloud by proxying or redirecting web traffic to the CP. This provides an added layer of protection on top of things such as antivirus to prevent malware from entering the enterprise via activities such as web browsing. In addition to protecting against malware, a cloud-based web security service might include usage policy enforcement, data backup, traffic control, and web access control.
- A CSP may provide a web-based e-mail service, for which security measures are needed. **E-mail security** provides control over inbound and outbound email, protecting the organization from phishing, malicious attachments, enforcing corporate polices such as acceptable use and spam prevention. The CSP may also incorporate digital signatures on all e-mail clients and provide optional e-mail encryption.
- **Security assessments** are third-part audits of cloud services. While this service is outside the province of the CSP, the CSP can provide tools and access points to facilitate various assessment activities.
- **Intrusion management** encompasses intrusion detection, prevention, and response. The core of this service is the implementation of intrusion detection systems (IDS) and intrusion prevention systems (IPS) at entry points to the cloud and on servers in the cloud. An IDS is a set of automated tools designed to detect unauthorized access to a host system. An IPS incorporates IDS functionality but also includes mechanisms designed to block traffic from intruders.
- **Security information and event management (SIEM)** aggregates (via push or pull mechanisms) log and event data from virtual and real networks, applications, and systems. This information is then correlated and analyzed to provide real-time reporting and alerting on information/events that may require intervention or other type of response. The CSP typically provides an integrated service that can put together information from a variety of sources both within the cloud and within the client enterprise network.
- **Encryption** is a pervasive service that can be provided for data at rest in the cloud, e-mail traffic, client-specific network management information, and identity information. Encryption services provided by the CSP involve a range of complex issues, including key management, how to implement virtual private network (VPN) services in the cloud, application encryption, and data content access.
- **Business continuity and disaster recovery** comprise measures and mechanisms to ensure operational resiliency in the event of any service interruptions. This is an area where the CSP, because of economies of scale, can offer obvious benefits to a cloud service client. The CSP can provide backup at multiple locations, with reliable failover

and disaster recovery facilities. This service must include a flexible infrastructure, redundancy of functions and hardware, monitored operations, geographically distributed data centers, and network survivability.

- **Network security** consists of security services that allocate access, distribute, monitor, and protect the underlying resource services. Services include perimeter and server firewalls and denial-of-service protection. Many of the other services listed in this section, including intrusion management, identity and access management, data loss protection, and web security, also contribute to the network security service.

Addressing Cloud Computer Security Concerns

Numerous documents have been developed to guide business thinking about the security issues associated with cloud computing. In addition to SP-800-144, which provides overall guidance, NIST has issued SP-800-146, *Cloud Computing Synopsis and Recommendations*, May 2012. NIST's recommendations systematically consider each of the major types of cloud services consumed by businesses including Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS). Security issues vary somewhat depending on the type of cloud service, but multiple NIST recommendations are independent of service type. Not surprisingly, NIST recommends selecting cloud providers that support strong encryption, have appropriate redundancy mechanisms in place, use authentication mechanisms, and offer subscribers sufficient visibility about mechanisms used to protect subscribers from other subscribers and the provider. SP-800-146 also lists the overall security controls that are relevant in a cloud computing environment and that must be assigned to the different cloud actors as listed in Table 16.4.

Technical	Operational	Management
Access control	Awareness and training	Certification, accreditation, and security Assessment
Audit and accountability	Configuration and management	Planning risk assessment
Identification and authentication	Contingency planning	System and services acquisition
System and communication protection	Incident response	
	Maintenance	
	Media protection	
	Physical and environmental protection	
	Personnel security	
	System and information integrity	

TABLE 16.4 Control Functions and Classes

As more businesses incorporate cloud services into their enterprise network infrastructures, cloud computing security will persist as an important issue. Examples of cloud computing security failures have the potential to have a chilling effect on business interest in cloud services, and this

is inspiring service providers to be serious about incorporating security mechanisms that will allay concerns of potential subscribers. Some service providers have moved their operations to Tier 4 data centers to address user concerns about availability and redundancy. Because so many businesses remain reluctant to embrace cloud computing in a big way, cloud service providers will have to continue to work hard to convince potential customers that computing support for core business processes and mission critical applications can be moved safely and securely to the cloud.

16.5 IoT Security

IoT is perhaps the most complex and undeveloped area of network security. To see this, consider Figure 16.8, which shows the main elements of interest for IoT security. At the center of the network are the application platforms, data storage servers, and network and security management systems. These central systems gather data from sensors, send control signals to actuators, and are responsible for managing the IoT devices and their communication networks. At the edge of the network are IoT-enabled devices, some of which are quite simple constrained devices and some of which are more intelligent unconstrained devices. As well, gateways may perform protocol conversion and other networking service on behalf of IoT devices.

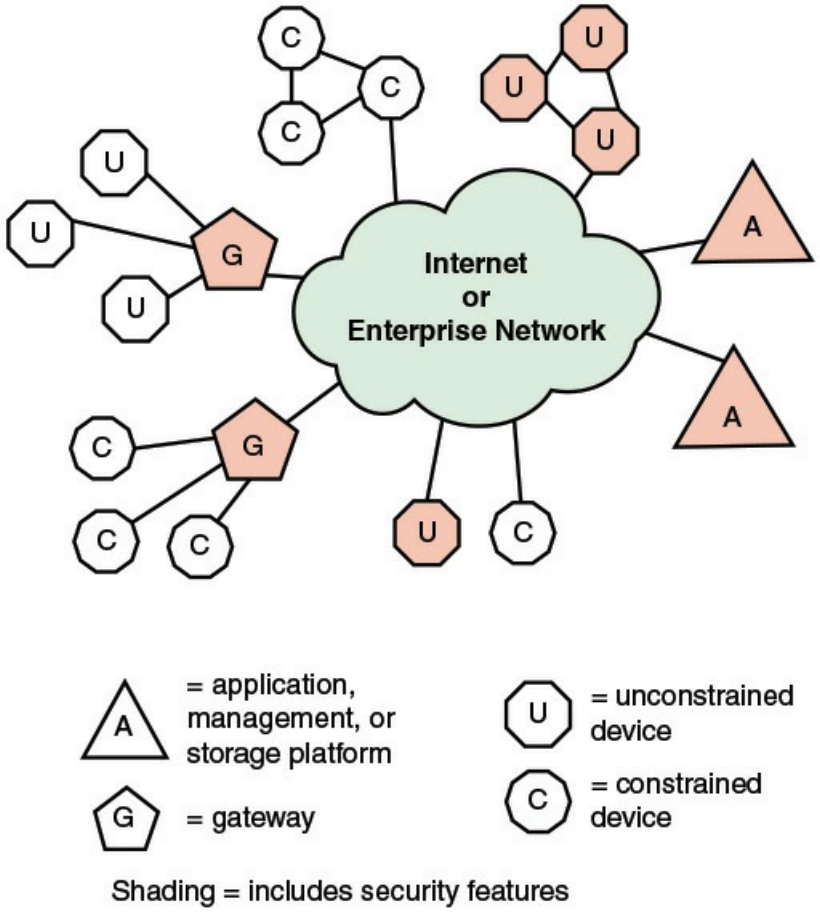

FIGURE 16.8 IoT Security: Elements of Interest

Figure 16.8 illustrates a number of typical scenarios for interconnection and the inclusion of security features.

The shading in Figure 16.8 indicates the systems that support at least some of these functions. Typically, gateways will implement secure functions, such as TLS and IPsec. Unconstrained devices may or may not implement some security capability. Constrained devices generally have limited or no security features. As suggested in the figure, gateway devices can provide secure communication between the gateway and the devices at the center, such as application platforms and management platforms. However, any constrained or unconstrained devices attached to the gateway are outside the zone of security established between the gateway and the central systems. As shown, unconstrained devices can communicate directly with the center and support security functions. However, constrained devices that are not connected to gateways have no secure communications with central devices.

The Patching Vulnerability

In an often-quoted 2014 article, security expert Bruce Schneier stated that we are at a crisis point with regard to the security of embedded systems, including IoT devices [SCHN14]. The embedded devices are riddled with vulnerabilities, and there is no good way to patch them. The chip manufacturers have strong incentives to produce their product with its firmware and software as quickly and cheaply as possible. The device manufacturers choose a chip based on price and features and do very little if anything to the chip software and firmware. Their focus is the functionality of the device itself. The end user may have no means of patching the system or, if so, little information about when and how to patch. The result is that the hundreds of millions of Internet-connected devices in the IoT are vulnerable to attack. This is certainly a problem with sensors, allowing attackers to insert false data into the network. It is potentially a graver threat with actuators, where the attacker can affect the operation of machinery and other devices.

IoT Security and Privacy Requirements Defined by ITU-T

ITU-T Recommendation Y.2066, *Common Requirements of the Internet of Things*, June 2014, includes a list of security requirements for the IoT. This list is a useful baseline for understanding the scope of security implementation needed for an IoT deployment. The requirements are defined as being the functional requirements during capturing, storing, transferring, aggregating and processing the data of things, as well as to the provision of services which involve things. These requirements are related to all the IoT actors. The requirements are as follows:

- **Communication security**: Secure, trusted, and privacy-protected communication capability is required, so that unauthorized access to the content of data can be prohibited, integrity of data can be guaranteed and privacy-related content of data can be protected during data transmission or transfer in IoT.
- **Data management security**: Secure, trusted, and privacy-protected data management capability is required, so that unauthorized access to the content of data can be prohibited, integrity of data can be guaranteed and privacy-related content of data can be protected when storing or processing data in IoT.

- **Service provision security**: Secure, trusted, and privacy-protected service provision capability is required, so that unauthorized access to service and fraudulent service provision can be prohibited and privacy information related to IoT users can be protected.
- **Integration of security policies and techniques**: The ability to integrate different security policies and techniques is required, so as to ensure a consistent security control over the variety of devices and user networks in IoT.
- **Mutual authentication and authorization**: Before a device (or an IoT user) can access the IoT, mutual authentication and authorization between the device (or the IoT user) and IoT is required to be performed according to predefined security policies.
- **Security audit**: Security audit is required to be supported in IoT. Any data access or attempt to access IoT applications are required to be fully transparent, traceable and reproducible according to appropriate regulation and laws. In particular, IoT is required to support security audit for data transmission, storage, processing, and application access.

A key element in providing security in an IoT deployment is the gateway. Y.2067, *Common Requirements and Capabilities of a Gateway for Internet of Things Applications*, June 2014, details specific security functions that the gateway should implement, some of which are illustrated in Figure 16.9. These consist of the following:

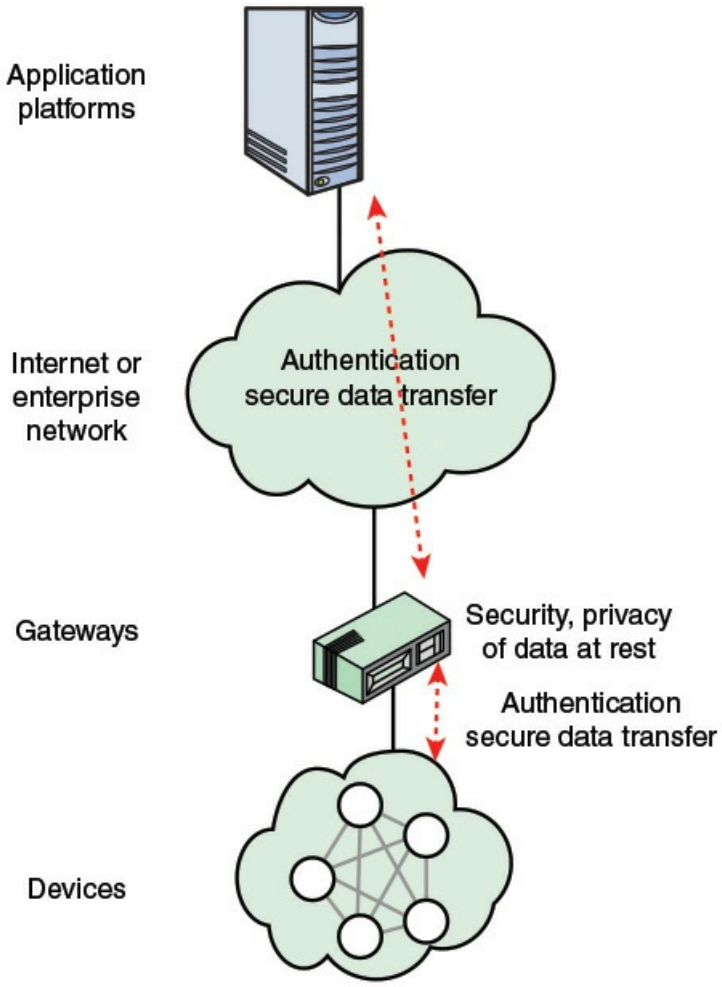

FIGURE 16.9 IoT Gateway Security Functions

- Support identification of each access to the connected devices.
- Support authentication with devices. Based on application requirements and device capabilities, it is required to support mutual or one-way authentication with devices. With one-way authentication, either the device authenticates itself to the gateway or the gateway authenticates itself to the device, but not both.
- Support mutual authentication with applications.
- Support the security of the data that are stored in devices and the gateway, or transferred between the gateway and devices, or transferred between the gateway and applications. Support the security of these data based on security levels.
- Support mechanisms to protect privacy for devices and the gateway.
- Support self-diagnosis and self-repair as well as remote maintenance.
- Support firmware and software update.
- Support auto configuration or configuration by applications. The gateway is required to support multiple configuration modes, for example, remote and local configuration, automatic and manual configuration, and dynamic configuration based on policies.

Some of these requirements may be difficult to achieve when they involve providing security services for constrained devices. For example, the gateway should support security of data stored in devices. Without encryption capability at the constrained device, this may be impractical to achieve.

Note that the Y.2067 requirements make a number of references to privacy requirements. Privacy is an area of growing concern with the widespread deployment of IoT-enabled things in homes, retail outlets, and vehicles and humans. As more things are interconnected, governments and private enterprises will collect massive amounts of data about individuals, including medical information, location and movement information, and application usage.

An IoT Security Framework

Cisco, which has played a lead role in the development of the IoT World Forum Reference Model (see Figure 15.4), has developed a framework for IoT security [FRAH15] that serves as a useful complement to the World Forum IoT Reference Model.

Figure 16.10 illustrates the security environment related to the logical structure of an IoT. The IoT model is a simplified version of the World Forum IoT Reference Model. It consists of the following levels:

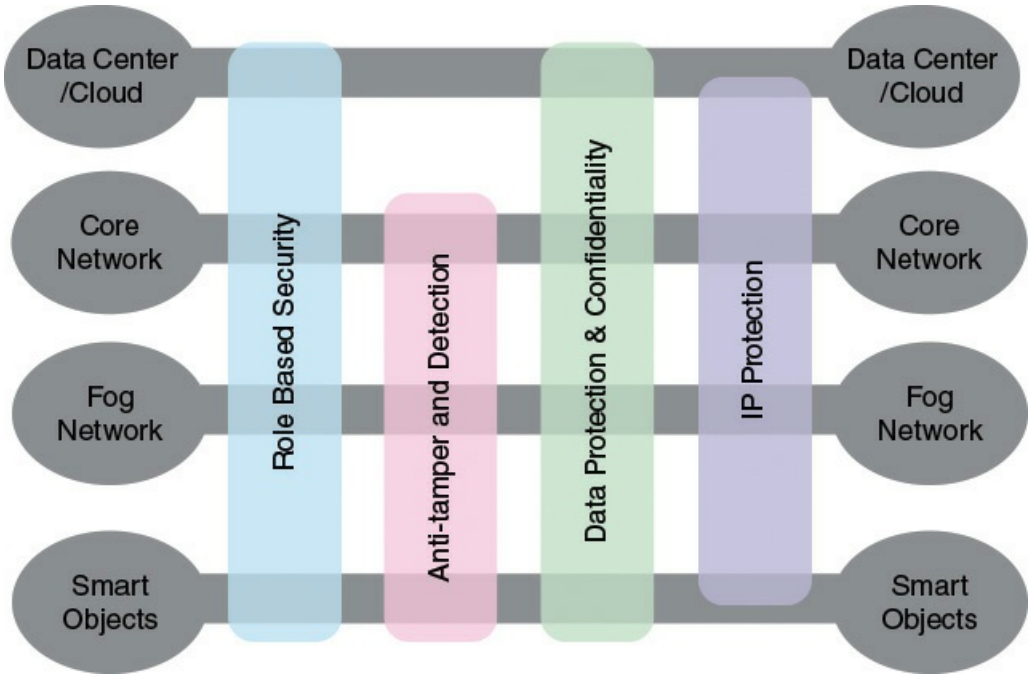

FIGURE 16.10 IoT Security Environment

- **Smart objects/embedded systems:** Consists of sensors, actuators, and other embedded systems at the edge of the network. This is the most vulnerable part of an IoT. The devices may not be in a physically secure environment and may need to function for years. Availability is certainly an issue. Also, network managers need to be concerned about the authenticity and integrity of the data generated by sensors and about protecting actuators and other smart devices from unauthorized use. Privacy and protection from eavesdropping may also be requirements.
- **Fog/edge network:** This level is concerned with the wired and wireless interconnection of IoT devices. In addition, a certain amount of data processing and consolidation may be done at this level. A key issue of concern is the wide variety of network technologies and protocols used by the various IoT devices and the need to develop and enforce a uniform security policy.
- **Core network:** The core network level provides data paths between network center platforms and the IoT devices. The security issues here are those confronted in traditional core networks. However, the vast number of endpoints to interact with and manage creates a substantial security burden.
- **Data center/cloud:** This level contains the application, data storage, and network management platforms. IoT does not introduce any new security issues at this level, other than the necessity of dealing with huge numbers of individual endpoints.

Within this four-level architecture, the Cisco model defines four general security capabilities that span multiple levels:

- **Role-based security:** RBAC systems assign access rights to roles instead of individual users. In turn, users are assigned to different roles, either statically or dynamically, according to their responsibilities. RBAC enjoys widespread commercial use in cloud and

enterprise security and is a well-understood tool that can be used to manage access to IoT devices and the data they generate.

- **Antitamper and detection:** This function is particularly important at the device and fog network levels but also extends to the core network level. All of these levels may involve components that are physically outside the area of the enterprise that is protected by physical security measures.
- **Data protection and confidentiality:** These functions extend to all level of the architecture.
- **Internet protocol protection:** Protection of data in motion from eavesdropping and snooping is essential between all levels.

Figure 16.10 maps specific security functional areas across the four layers of the IoT model. A 2015 Cisco White Paper on IoT security [FRAH15] also proposes a secure IoT framework that defines the components of a security facility for an IoT that encompasses all the levels, as shown in Figure 16.11, and described in the list that follows.

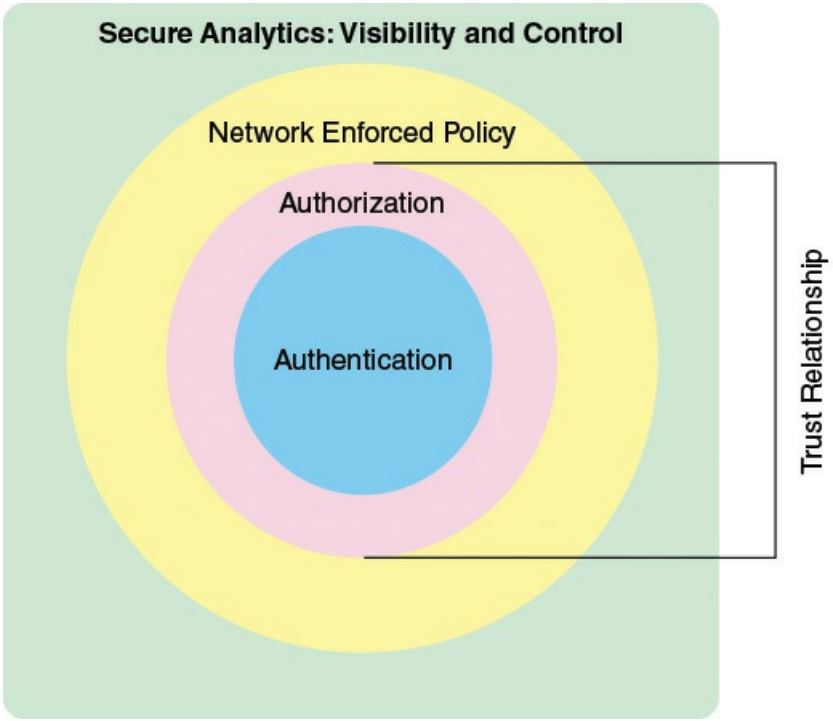

FIGURE 16.11 Secure IoT Framework

- **Authentication:** Encompasses the elements that initiate the determination of access by first identifying the IoT devices. In contrast to typical enterprise network devices, which may be identified by a human credential (for example, username and password or token), the IoT endpoints must be fingerprinted by means that do not require human interaction. Such identifiers include RFID, x.509 certificates, or the MAC address of the endpoint.
- **Authorization:** Controls a device's access throughout the network fabric. This element encompasses access control. Together with the authentication layer, it establishes the necessary parameters to enable the exchange of information between devices and between devices and application platforms and enables IoT-related services to be performed.

- **Network enforced policy:** Encompasses all elements that route and transport endpoint traffic securely over the infrastructure, whether control, management or actual data traffic.
- **Secure analytics, including visibility and control:** This component includes all the functions required for central management of IoT devices. This involves, first, visibility of IoT devices, which simply means that central management services are securely aware of the distributed IoT device collection, including identity and attributes of each device. Building on this visibility is the ability to exert control, including configuration, patch updates, and threat countermeasures.

An important concept related to this framework is that of trust relationship. In this context, trust relationship refers to the ability of the two partners to an exchange to have confidence in the identity and access rights of the other. The authentication component of the trust framework provides a basic level of trust, which is expanded with the authorization component. The Cisco IoT security white paper [FRAH15] gives the example that a car may establish a trust relationship with another car from the same vendor. That trust relationship, however, may only allow cars to exchange their safety capabilities. When a trusted relationship is established between the same car and its dealer's network, the car may be allowed to share additional information such as its odometer reading and last maintenance record.

Conclusion

Computer and network security protocols, technologies, and policies have developed and matured over the past decades, tailored to the needs of enterprises, governments, and other users. Although there is an ongoing arms race between attackers and defenders, it is possible to build a powerful security facility for traditional networks and for SDN/NFV networks. The sudden explosion of IoT networks with millions to billions of devices poses an unprecedented security challenge. A model and framework such as that of Figures 16.10 and 16.11 can serve as a foundation for the design and implementation of an IoT security facility.

16.6 Key Terms

After completing this chapter, you should be able to define the following terms.

accountability

attack surface

authenticity

availability

confidentiality

data confidentiality

data integrity

hypervisor introspection

integrity

privacy

role-based access control (RBAC)

Security as a Service (SecaaS)

system integrity

Transport Layer Security (TLS)

16.7 References

CSA11: Cloud Security Alliance. *Security as a Service (SecaaS).* CSA Report, 2011.

CSA13: Cloud Security Alliance. *The Notorious Nine Cloud Computing Top Threats in 2013.* CSA Report, February 2013.

HAWI14: Hawilo, H., et al. "NFV: State of the Art, Challenges, and Implementation in Next Generation Mobile Networks." *IEEE Network*, November/December 2014.

HOGG14: Hogg, S. "SDN Security Attack Vectors and SDN Hardening." *Network World*, Oct 28, 2014.

NAKI15: Nakina Systems. *Achieving Security Integrity in Service Provider NFV Environments.* Nakina Systems white paper, 2015.

STAL15: Stallings, W., and Brown, L. *Computer Security: Principles and Practice.* Englewood Cliffs, NJ: Pearson, 2015.

Chapter 17. The Impact of the New Networking on IT Careers

> You don't understand! I coulda had class. I coulda been a contender. I could've been somebody, instead of a bum, which is what I am.
>
> —Marlon Brando, *On the Waterfront*, 1954

Chapter Objectives: **After studying this chapter, you should be able to**

- Discuss the changing responsibilities of network professionals and the impact on job positions.
- Present an overview of DevOps.
- Understand the role of DevOps for implementing networking systems.
- Understand the relevance of training and certification programs.

The network landscape is changing rapidly in a variety of ways and in a number of directions. To further their careers, network professionals need to not only master new technical skills but also broaden the scope of their involvement in the many facets of network technology, management, and deployment. This chapter aims to provide some guidance and information that will be useful in protecting and enhancing your career prospects in the new networking landscape.

The chapter begins with some overall thoughts about the changing roles of network professionals. Next, the chapter focuses on one specific area that might be overlooked in developing your career-building skills: DevOps. This is followed by a discussion of training and certification. The chapter closes with a description of online resources that can be an ongoing source of information and support.

17.1 The Changing Role of Network Professionals

The emerging networking era has many ramifications that the alert network professional should consider. We mention some here:

- The network infrastructure is unlikely to be sourced from a single vendor. The infrastructure has multiple layers, defined interfaces (both horizontal and vertical), reliance on abstraction, and a mix of local and cloud/fog-based elements.
- Application workloads are changing, both in the variety and pace. Software modules that manage, utilize, and even define the network infrastructure need to be incorporated into the network software environment.
- The available toolset of the network professional is proliferating rapidly, including languages, scripting tools, and a growing variety of packaged products to help do network design, deployment, operations, management, and security. IT executives are aware of these tools and expect their networking team to use them.
- Network functions are increasingly being defined, implemented, and managed using

software techniques, such as software-defined networking (SDN) and network functions virtualization (NFV). This “soft” nature of networks compels an increasingly collaborative approach to IT management and to network development and operations.

Network professionals cannot expect to move forward with skills learned in college or training so far obtained. SDN and NFV open up the network ecosystem to many more players so that the complex world of networking is accessible to people coming from a variety of backgrounds. Networking roles and responsibilities will be in flux, with job slots disappearing and opening up. Networking professionals need to seize opportunities for both in-house and third party training to maintain their competitive edge.

Changing Responsibilities

A Webtorials paper by Metzler [METZ14b] lists the following as key characteristics of the emerging role of network and IT infrastructure professionals:

- **More emphasis on programming:** At minimum, the proliferation of application programming interfaces (APIs) as part of the SDN and NFV network structure requires senior level IT professionals to have some level of understanding of programming to better interact with enterprise software development units. And organizations may want to leverage the API functionality newly available by having networking professionals write programs that utilize those APIs. This is discussed further in Section 17.2.
- **An increased knowledge of other IT disciplines:** IT will become less separated by specific areas of expertise (storage, networking, virtualization, and security) and more cross-functional as teams interoperate with each other. The increased emphasis on collaboration and DevOps (discussed in the next section) requires an amalgamation of skills spanning everything from IT security to database design and application architecture, plus everything in between. While each individual on the team has a particular strength, each one also needs to have working knowledge in other areas.
- **Heightened emphasis on security:** Security expertise becomes more critical as data is secured on premises, in the cloud, and on user devices. Data is the lifeblood of any company, and so determining and enforcing policies that keep things secure without impacting users’ ability to get their work accomplished will be critical.
- **More focus on setting policy:** SDN and NFV enable IT organizations to implement a policy-driven infrastructure in a more dynamic and granular fashion than was previously possible.
- **More knowledge of the business:** SDN, NFV, and QoE provide management with the technology base for providing an agile response to business needs and customer requirements. New application software is generated to run on the network and the virtualized network elements are modified and repositioned rapidly to adjust to enterprise and user needs. This places the burden on the network professional to understand how the network is to be managed and configured to support this dynamic environment. Another consideration is that the ability of the IT organization to justify an investment in IT is increasingly tied to the ability of the organization to concretely demonstrate the business value of that investment.

- **More understanding of applications:** Cloud computing and IoT open up the range of applications that need to be supported on networks. The architecture of these applications is broadening as well, with simple client/server models supplanted or enhanced by applications structures that spread out vertically (mutitier) and horizontally (peer cooperation). Complex applications, such as customer relationship management (CRM), actually consist of several modules, with a range of network requirements. IT infrastructure and network professionals in particular need to better understand these new architectures and complex applications to ensure that the emerging set of technologies are designed and architected appropriately.

Another interesting take on what is needed to succeed in the new networking environment is provided in a paper by Pretz [PRET14], which lists the following five skills needed by network professionals:

- The ability to incorporate know-how from the IT and network domains, which have grown independently of each other over the years but are now converging.
- An understanding of industrial mathematics, a branch of applied mathematics. Those with this knowledge will be better able to understand technical issues, formulate precise and accurate mathematical models, and implement solutions using the latest computer techniques. An understanding of this field will help in developing systems by applying machine learning and cognitive algorithms, which are expected to lessen the complexity and dynamic nature of SDNs.
- A mastery of software architecture and open source software, which is needed to develop SDN tools and applications. It will also be helpful to understand software verification and validation processes, which ensure that software meets specifications and fulfills its intended purpose. Some engineers assume they'll need programming skills, but that's not necessarily so, because software applications for SDNs from third parties are already available.
- A background in big data analytics to understand how to handle the huge amounts of data expected from SDNs. Someone skilled in big data analytics will not only be able to manage more data but also know the right questions to ask should problems arise. Such analytics will also help engineers make smart, data-driven decisions.
- Expertise in cybersecurity, because security must be everywhere within SDNs. It needs to be built into the architecture and also must be delivered as a service to protect the availability, integrity, and privacy of connected resources and information.

Impact on Job Positions

In a Global Knowledge white paper, Hales [HALE14] lists the following as likely impacts of SDN and NFV on individual job positions:

- **Network administrator:** Those with the skills to design and manage software-dominated networks and to plan migration strategy from the existing environment will be in high demand.
- **Virtualization administrator:** Administrators with more advanced skills will be needed

for figuring out how to implement cloud systems and make them work within an existing infrastructure. Virtualization administrators will need to work more closely with the storage, network, security, and application teams to make them seamlessly work together.

- **Applications administrator:** Applications administrators need to be aware of the many implications of SDN and NFV APIs on applications. This includes the fact that applications can request the network to provide them with the bandwidth and latency needed for the application to work correctly. Administrators will need to know what these requirements are and work with other application administrators to ensure that the needs of all applications are being met. Security needs will also change in unanticipated ways so that the applications administrator needs a good understanding of security services and mechanisms.
- **Security administrator:** The security administrator will likely need to work more closely with other types of administrators to ensure that appropriate policies and rules are designed, enforced, and audited to ensure compliance. As companies move more to the cloud and encourage users to bring your own device (BYOD), the need for this type of administrator will only increase.
- **Developer:** Developers might be used to integrate functionality into the APIs provided by the SDN and NFV controllers, or they may write applications that can make requests of the network. This may require additional general networking knowledge as well as knowledge of the specific APIs that need to be used for a given problem. Developers will need to think about security in greater detail and pass on security requirements to security, application, virtualization, and/or network teams to ensure that the requirements of the application are met and to modify the application as needed.
- **IT manager:** IT managers must become generalists with an ability to understand the new networking capabilities, the security demands of the new environment, and the need to integrate application development with network development. A collaborative mindset, such as is required by DevOps (discussed in Section 17.2), must be cultivated by all within the organization.

Bottom Line

The need for a strong staff of networking professionals is not going to go away, no matter how many automated tools are added to the new networking infrastructures. But the roles, responsibilities, and skills needed to thrive in the new networking environment are changing.

17.2 DevOps

A review of technical and management literature on SDN, NFV, cloud, and Internet of Things (IoT) shows a frequent reference to the need for personnel who understand and can use a DevOps approach to designing, installing, and managing these new network technologies. This section first provides an overview of the concept of DevOps and then looks at how it applies to modern networking technologies.

DevOps Fundamentals

In just a few short years, **DevOps** has gone from a buzzword to an accepted method of software development and deployment. Enterprises large and small are trying to get a grasp on what DevOps is and what impact it can have on their organizations. The attention is coming not just from IT executives and CIOs but also from business managers who are beginning to recognize the potential of DevOps to enable business units to become more efficient, deliver higher-quality product, and be more agile and innovative. Major software organizations, including IBM and Microsoft, are rapidly expanding their DevOps offerings.

The focus of DevOps has been the development of application software and support software. The essence of the DevOps philosophy is that all participants in creating a product or system should collaborate from the beginning, including business unit managers, developers, operations staff, security staff, and end-user groups.

To understand the DevOps approach, we need to briefly outline the typical stages in the development and deployment of applications. As described in *Application Release and Deployment for Dummies*, most application vendors and in-house application developers follow a lifecycle similar to the following [MINI14]:

- **Development (DEV):** Developers build and deploy code in a test environment, and the development team tests the application at the most basic level. When the application meets certain criteria for advancement, it moves to SIT.
- **System integration testing (SIT):** The application is tested to ensure that it works with existing applications and systems. When the application meets the criteria of this environment, it is deployed to UAT.
- **User acceptance testing (UAT):** The application is tested to ensure that it provides the required features for end users. This environment usually is production-like. When the application passes these requirements, it moves to production.
- **Production (PROD):** The application is made available to users. Feedback is captured by monitoring the application's availability and functionality. Any updates or patches are introduced in the DEV environment and follow the same cycle.

Traditionally, an information system development project proceeds sequentially through these stages, without delivering working pieces in between and without obtaining customer feedback on the way. The entire process is called **waterfall development**. With such large projects, once each stage is completed, it cannot be easily reversed, much like trying to move up a waterfall. Beginning in the early 2000s, **agile software development** began to gain favor. The agile methods emphasize teamwork, customer involvement and, most significantly, the creation of small or partial pieces of the total system that are tested in a user environment. For example, an application with 25 features might be prototyped with only 5 or 6 thoroughly completed before adding more, and so on. Agile development has proven to be more effective in dealing with changing requirements during the development phase, which always seem to occur.

Agile development is characterized by frequent releases, in an iterated loop fashion, with a certain amount of automation in the form of tools that can be used to support collaboration. DevOps takes this philosophy much further. It is characterized by rapid releases, feedback loops

embedded throughout the process, and a comprehensive set of tools and documented best practices to automate the DevOps process.

Figure 17.1, from *DevOps for Dummies*, provides an overview of the DevOps process [SHAR15].

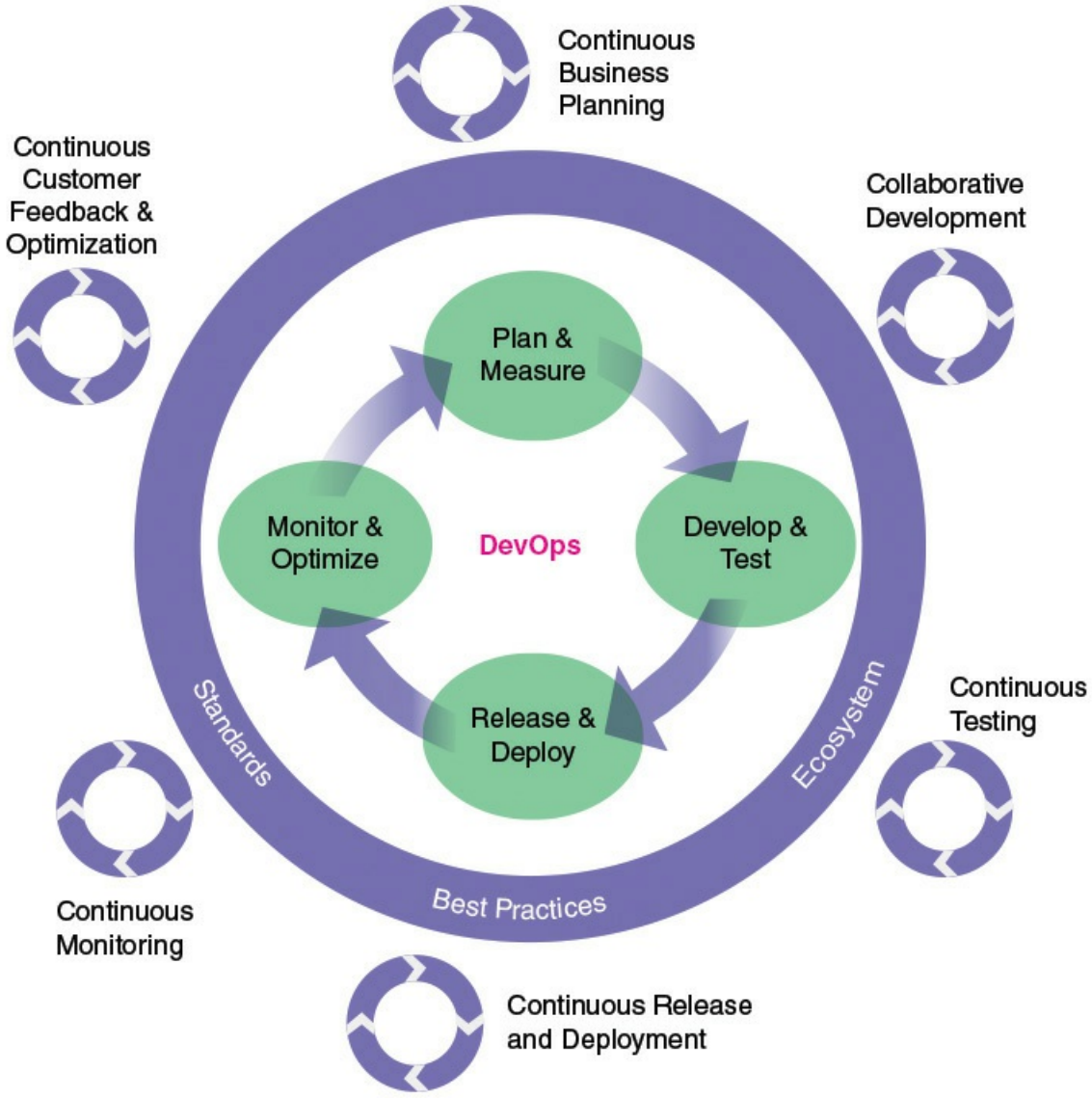

FIGURE 17.1 DevOps Reference Architecture

DevOps can be viewed as a repetitive cycle of four major activities:

- **Plan and measure:** Focuses on business units and their planning process. The planning process relates business needs to the outcomes of the development process. This activity can start with small, limited portions of the overall plan, identifying outcomes and resources needed to develop the required software. The plan must include developing measures that are used to evaluate software, adapt and adjust continually, relate to customer needs, and continually update the development plan and the measurement plan. The measurement function can also be applied to the DevOps process itself to ensure that

the right automated tools are being used and that collaboration is ongoing.

- **Develop and test:** Focuses on collaborative development, continuous integration of new code, and continuous testing. It focuses on streamlining development and testing teams' capabilities. Useful tools are automated tracking of testing against measured outcomes and virtualized test beds that enable testing in an isolated but real-world environment.
- **Release and deploy:** Provide a continuous delivery pipeline that automates deployment to test and production environments. Releases are managed centrally in a collaborative environment that leverages automation. Deployments and middleware configurations are automated and then mature to a self-service model that provides individual developers, teams, testers, and deployment managers with a capability to continuously build, provision, deploy, test, and promote. Infrastructure and middleware provisioning evolves to an automated then self-service capability similar to application deployment. Operations engineers cease manually changing environments; instead, they focus on optimizing the automation.
- **Monitor and optimize:** Includes the practices of continuous monitoring, customer feedback, and optimization to monitor how applications are performing post-release, allowing businesses to adapt their requirements as needed. Customer experience is monitored to optimize experiences within business applications. Optimization to customer key performance indicators that reflect business value attainment is part of the continuous improvement program.

Figure 17.2, from the Microsoft white paper *Enterprise DevOps* [MICR15], provides another useful perspective on DevOps. DevOps is intended to improve the efficiency and effectiveness of the process of managing applications throughout their lifecycle. With the introduction of agile software development, organizations have developed **application lifecycle management (ALM)** practices to integrate the business, development, QA, and operations functions in a virtuous cycle for greater agility in delivering continuous value.

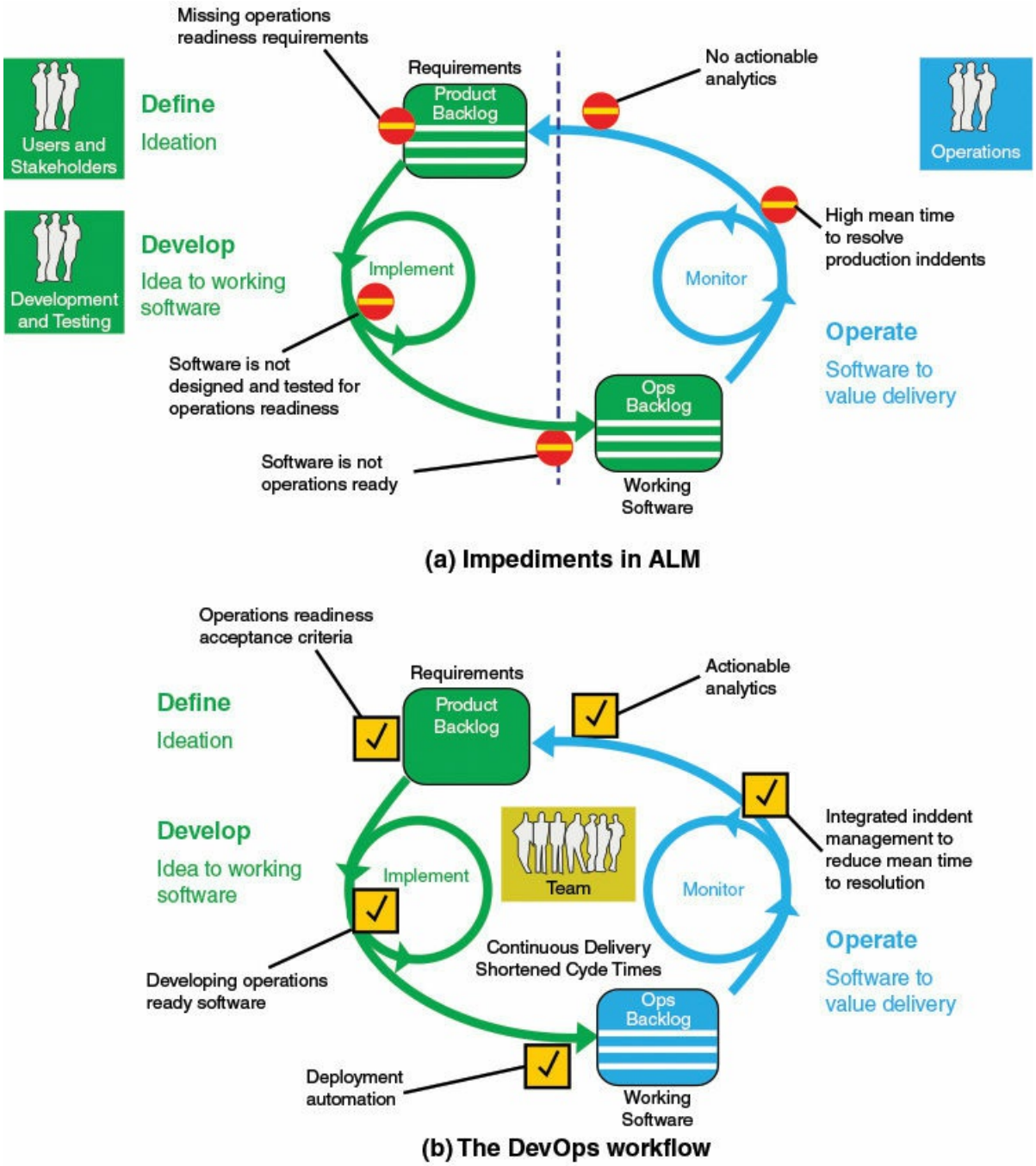

FIGURE 17.2 Modern Application Lifestyle Management

As part a of Figure 17.2 shows, ALM practices, as they have developed, have encountered a number of impediments to agile and effective delivery of the final product. These arise from the conventional divide that exists between development and operations functions. A key theme illustrated here is the danger that operations requirements are being de-prioritized to accommodate functional needs. DevOps intends to address these impediments, as shown in part b of Figure 17.2.

Fundamentally, DevOps rests on two key foundations: collaboration and automation.

Collaboration begins with management policy to encourage and require the various actors in the software development and deployment process to work together. Automation consists of tools that support that collaboration and are designed to automate as much as possible the cyclic process illustrated in Figures 17.1 and 17.2.

A number of companies now offer DevOps automation tools. For example, in 2014 Microsoft introduced a number of tools that work as part of their Visual Studio offerings. Visual Studio is a set of developer tools and services to assist users in creating apps on Microsoft platforms and in the cloud. One of the additions is releasing management software that automates many of the chores that are needed to be done to move a software program from development to production, such as alerting the appropriate managers, and preparing the production server to run the software. Another DevOps-minded feature Microsoft has introduced for Visual Studio is called Cloud Deployment Projects, which allows organizations to capture and reuse the configuration settings of new applications, in order to speed the deployment times. The configuration settings, or blueprints, are captured within a virtual machine (VM), which then can be deployed, holding the application, in the Microsoft Azure cloud. Microsoft also introduced its Application Insights software. Application Insights provides a way to instrument an application so the developers can determine if it is working correctly, and how people are using the software program. This could help developers pinpoint bugs, as well as get early insight into behavioral issues, such as a sudden fall-off of use due to a bad redesign.

The Demand for DevOps

IT departments are increasingly relying on DevOps. For example, a recent report [DICE15] on the job-listing site Dice stated that "senior system administrators with DevOps and an engineering background are in the right area of their careers. In markets like Silicon Valley, recruiting DevOps talent can be a headache. It's not unusual for multiple offers, counteroffers, and rising salaries for DevOps experience." Table 15.1 shows the number of active job listings for DevOps engineers, managers, architects, and so on within a 100-mile radius of six U.S. cities. DevOps clearly has "arrived" as a skill set that tech employers are seeking.

City	Number of Listings
Boston	106
New York	183
Washington, D.C.	109
Chicago	53
San Francisco	319
Dallas	85

TABLE 17.1 Recent DevOps Job Listings on Dice by Location (May 2015)

DevOps for Networking

Although DevOps was created and has evolved to support the application development and deployment process, it can also be applied in the networking context. This is because the

networking infrastructure is increasingly software defined and software driven:

- **Software-defined networking (SDN):** SDN defines network behavior in software. Utilities and apps at the control and application level build on the basic capability provided by the split between the control and data planes. Network designers and network managers need to be able to rapidly respond to changing network conditions and requirements and the need for new customer-driven applications.
- **Network functions virtualization (NFV):** NFV defines the structure and functioning of the network in software, with the deployment of virtual compute, storage, and network functions. The NFV software environment is complex, involving the interaction of a host virtual network functions (VNFs) and management and operations software. This is an environment that requires rapid response to changing conditions and demands.
- **QoS/QoE:** The demands of quality of service (QoS) and especially quality of experience (QoE) dictate a process that is driven by end-user analytics and the is best served by a rapid development and deployment cycle to ensure that the network is responsive to the end user's needs.
- **Cloud:** Whether it be IaaS, PaaS, or SaaS, and whether it be public or private cloud, cloud managers and providers have a constant, ongoing need to modify and enhance cloud offerings. To meet user expectations, this must be done in a highly agile manner.
- **Internet of Things (IoT):** Although IoT involves lots of physical "stuff," the overall architecture from the fog computing edge to the central application platforms, requires rapid response to changing conditions to provide expected performance, as well as the need to constantly upgrade and modify the networks to deal with a rapidly changing mix of IoT devices.

In a nutshell, a DevOps approach is not just for applications, web server software, and the like; it is also for network infrastructure. For network managers and network engineers who are designing and deploying network infrastructure software and who are modifying the network infrastructure on demand, the DevOps approach can involve a number of aspects, including the following:

- Increased collaboration with network operations staff so as to anticipate how network changes will impact day-to-day operations, developing metrics for measuring the impact of changes, and developing procedures for creating a back-and-forth between development and operations.
- Examining the software and network infrastructure deployment pipeline with a focus on the processes that govern pipeline flow to determine how to enhance efficiency and remove impediments.
- Adopting automation tools to eliminate repetitious tasks.

All the networking technologies discussed in this book lend themselves to the DevOps approach. But perhaps the most prominent area is cloud computing/networking, where providers seem to be ahead of the curve in employing DevOps techniques. As pointed out in the Dice report, *Why DevOps Is CPR for Cloud Applications*, [DICE13], "The cloud lends itself naturally to DevOps in that it's heavily driven by APIs and frameworks that can easily be incorporated into automated, DevOps processes. It is the API-driven, self-service provisioning that makes the

cloud the cloud, so DevOps is a natural fit where clouds are involved. That means a good way to succeed or move into a position in the cloud is to polish your scripting and API-targeting skills. Being able to demonstrate experience with public cloud provider APIs or private cloud management frameworks will go a long way toward building a portfolio of cloud skills that will make you more attractive to prospective (and current) employers."

An *Information Week* article [MACV15] points out a common concern among network engineers as the beating of the DevOps drum gets louder is the associated focus on programmability. In particular, engineers are concerned they may be required to, well, code (rightly so, given phrases like *infrastructure as code*). They are concerned about the skills and skill sets they need that they may not have. Two things need to be said about that. First, the type of coding likely to be involved is scripting, rather than large software development with C, C++, Java, and so on. Network engineers use tools such as Python, Perl, Bash, and Curl to script common tasks across a variety of devices. To move this scripting approach into the DevOps realm, the network engineer needs to learn some tools that are integral to a networking DevOps environment.

One such tool is a version control system, such as Git. In addition to being a repository for software source code, version control systems can hold configuration data for infrastructure such as routers, firewalls, switches, and Apache web servers. Maintaining configuration data in a version control system provides an element of change control. It allows you to track things such as when a firewall rule was introduced or when an Apache vhost was added. The scripts written for device tasks (for example, in Python) can be stored in Git, where they are versioned and controlled. Further, with Git, scripting can be used to automate much of the task of populating the version control data. In addition, configuration management tools, such as Puppet or Chef, can be used to generate templates that are stored in Git.

A second point to make is that DevOps for networking is broader than just scripting, such as working with relevant staff to optimize processes and manage the infrastructure in a collaborative fashion that takes into account development, operations, and user needs. Another ongoing task is determining what (and how) you need to measure to meet the business priorities that are driving DevOps in the first place: faster time to market, reduced risk, and lower costs.

Even so, the mastery of skills with particular software tools and packages is a good way to build DevOps credibility. The Dice report *Critical Skills for DevOps Engineers*, [DICE14] lists the following as the four main clusters of skills and tools to succeed in a DevOps role:

- **Puppet, Chef, Vagrant, CFEngine, and Bcfg2:** Maintaining consistent system performance is critical. This means being up and available, as well as fast and reliable. Experience with these configuration management tools will help you manage software and system changes repeatedly and predictably.
- **Jenkins, Maven, Ant, CruiseControl, and Hudson:** A key part of your job is making it faster and easier to create and deploy software. Experience with tools like these will help ensure you have what you need to keep things moving.
- **Git, SVN, CVS, Visual Studio Online, and Perforce:** Version control is important to DevOps so developers don't get in each other's way. Use of these source control systems allows for collaboration on software projects and makes it easy to manage changes and updates.
- **Nagios, Munin, Zabbix, Sensu, LogStash, CloudWatch, Splunk, and NewRelic:** As a

DevOps professional, you must always keep tabs on performance. While the specifics of each tool are different, you should know the philosophy and principles behind each of them so that you can implement them effectively.

Strong experience with one technology in a cluster usually translates well to the others with only a few weeks of training, and because many of these tools are relatively new, you should be willing and able to apply your existing knowledge to new tools as needed for the role.

DevOps Network Offerings

A good indication of the growing awareness of the need for DevOps for modern network providers is found in the most recent annual NVF report from SDxCentral (SDxCentral Network Functions Virtualization Report, 2015 Edition). The following companies are listed as providing DevOps-related products:

- **Brocade Mobile Analytics:** Provides a full mobile network visibility capability stack. Modular product architecture lends itself to DevOps model for rapidly deploying custom solutions that fulfill mobile operators' unique needs.
- **Red Hat Enterprise Linux Atomic Host:** An NFV software platform. It includes tools to enable IT organizations to quickly realize the benefits of DevOps practices, including faster delivery of features and continual improvement.
- **SuperCloud:** A vendor-neutral NFV services orchestration platform. Enables data center and cloud service providers to deploy and manage VNF and SDN applications. Designed from DevOps and service automation mindset to fulfill the needs of network administrators that support IT application developers.
- **CloudShell:** A DevOps-oriented cloud management platform that provides self-service access to complex network environments comprised of bare metal as well as virtualized components. CloudShell is used to automate DevOps labs and data centers for development, testing, training, support, proof of concept, and open communities. CloudShell markets itself as the leading automation platform for network DevOps.

The list of NFV- and SDN-related vendors whose offerings reflect or support DevOps is likely to grow dramatically in the next few years.

Cisco DevNet

In 2015, Cisco announced a new approach, known as DevNet for Cisco customers and partners for employing DevOps. DevNet is meant to be a community of the enterprise network developers among its customers, its independent software vendors, independent systems integrators, and Cisco partners, producing software applications to run the programmable network of the future.

Cisco DevNet provides software developer kits (SDKs), visual modeling tools, ready-to-use code samples, and more accessible REST-based APIs through partner Mulesoft. Also, DevNet is a community where members may come to rely on each other for shared experience and support. In addition, DevNet will serve as an education and delivery vehicle for the Cisco approach to

SDN, its Application Centric Infrastructure.

Conclusion on the Current State of DevOps

This chapter devotes considerable space to DevOps for two reasons. First, DevOps will become increasingly critical for managing the stupendously complex networks that can be deployed with technologies such as NFV and SDN. Second, it is perhaps less obvious to the career-minded individual that DevOps know-how is a key skill than is the need for an understanding of SDN, NFV, QoE, and so on. The employee or job seeker with DevOps skills will have a competitive advantage going forward.

17.3 Training and Certification

The technologies discussed in this book are rapidly coming to dominate the networking industry and both private sector and government users. Networking professionals who have got this far in the book should by now be convinced of the need to learn these technologies and demonstrate competence in them. With all the changes taking place, experts warn networking pros that they will be left behind if they do not add new skills. Training and certification are the ideal vehicles for this. In a 2013 survey [BORT13] of 700 network professionals, Some 60 percent said a certification led to a new job; 50 percent said they earned more pay, with 40 percent saying their pay increased by more than 10 percent directly because of a certification; and 29 percent said a certification led to a promotion.

Fortunately, there is a large and growing number of opportunities to learn how to use the new networking technologies through certification programs.

Certification Programs

Tables 17.2 through 17.4 show some of the certification programs available related to SDN, network virtualization, and the cloud, respectively. Many of these emphasize the products of the companies that offer the training and certification, and so networking professionals can choose programs that either enhance their skills for their current position or for positions they want to seek. As for IoT, there are few offerings from the traditional sources. One recently introduced offering is the Cisco Industrial Networking Specialist Certificate. This training and certification program is for information technology (IT) and operational technology (OT) professionals in the manufacturing, process control, and oil and gas industries, who will be involved with the implementation, operation, and support of networked industrial products and solutions. We can expect to see many more such offerings.

Table 17.5 lists a number of other certification offerings in networking-related fields.

Type of Certification	Certification	Description
Virtualization Certification	VMware Certified Associate - Data Center Virtualization (VCA-DCV)	Enables IT pros to have greater credibility when discussing data center virtualization, and how to virtualize the data center with vSphere.
	VMware Certified Professional 5 - Data Center Virtualization (VCP5-DCV)	Designed to enable IT pros to effectively install, deploy, scale, and manage VMware vSphere environments, and also provides skills obtained from a minimum of 6 months experience with VMware infrastructure technologies. Candidates must complete a VMware-authorized training course and hands-on experience with VMware technologies.
	VMware Certified Advanced Professional 5 - Data Center Administration (VCAP5-DCA)	To earn this certification, IT pros must complete a VMware-authorized training course and conduct hands-on experience with VMware technologies. Candidates receive the knowledge required to effectively install, deploy, scale, and manage VMware vSphere environments, as well as the skills gained by a minimum of 6 months experience with VMware infrastructure technologies.

VMware Certified Advanced Professional 5 - Data Center Design (VCAP5-DCD)	Requires candidates go through a lab-based exam, performing tasks using actual equipment to verify skills at installing, configuring, and administering large and complex virtualized environments. Receiving this credential gives an IT pro an advanced certificate that demonstrates expertise with VMware vSphere 5, as well as the ability to use automation tools and implement virtualized environments.
VMware Certified Design Expert 5 - Data Center Virtualization (VCDX5-DCV)	Reserved for top design architects highly skilled in VMware enterprise deployments. This certification program is designed for veteran professionals who want to validate and demonstrate their expertise in VMware technology. The certification is completed through a design-defense process, where all candidates must submit and effectively secure a production-ready VMware solution before a board of veteran VCDX-DCV holders.
Citrix Certified Associate - Virtualization (CCA-V)	Validates the skills and knowledge of IT operators and administrators to manage, maintain, monitor and troubleshoot XenDesktop 7 solutions.
Citrix Certified Professional - Virtualization (CCP-V)	Validates the skills and knowledge of experienced IT solution builders, such as engineers and consultants, to install, configure, and roll out common XenDesktop 7 solutions.

	Citrix Certified Expert - Virtualization (CCE-V)	Recognizes the skills and knowledge of experienced IT solution designers, such as architects, engineers, and consultants, to assess and design comprehensive XenDesktop 7 solutions.
Networking Certification	Cisco Entry Level Certification	Designed for those who have a keen interest in networking and want to start a career in the same. This certification serves as a stepping stone.
	Cisco Associate Level Certification	Designed for candidates who already have an entry-level certification or some experience in networking, such as troubleshooting or network design. These certifications would serve as the foundation to a career in networking.
	Cisco Professional Level Certification	For networking pros who have significant experience and skills in networking domains and are ready to go to the next level. These certifications are ideal for candidates wanting to explore new avenues within networking, with varying roles and responsibilities.
	Cisco Expert Level Certification	For networking pros who are recognized for their expert network engineering skills and mastery of Cisco products and solutions. These certifications are designed for those who want deep technical networking knowledge and challenging assignments.

	Cisco Architect Certification	Recognizing deep technical expertise, these certifications are the highest level Cisco has to offer.
	Juniper Service Provider Routing and Switching	This track is designed for those who have been working with infrastructure or access products in a Juniper routing/switching end-to-end environment, predominantly within the telecommunications arena or a Fortune 100 enterprise environment.
	Juniper Enterprise Routing and Switching	Designed for candidates working in small to large enterprise settings that install and support Juniper-based networks in which LAN and WAN routers and switches reside.
	Juniper Junos Security	This track is directed toward those who design and implement Juniper secure networks.
Project Management Certification	Certified Associate in Project Management (CAPM)	Project Management Institute (PMI) certification designed for candidates who are less experienced project practitioners looking to demonstrate their commitment to project management and want to improve their ability to manage larger projects and earn additional responsibilities.

PMI Agile Certified Practitioner (PMI-ACP)	Designed for candidates who are active users of agile practices in their organization or who are in the process of adopting agile methods, this certification is ideal for demonstrating commitment to this rapidly growing approach to project management.
Project Management Professional (PMP)	Designed for those who want to demonstrate competence in leading and directing project teams. The ideal candidate is an experienced project manager looking to solidify skills, stand out to employers, and maximize their earning potential.
Portfolio Management Professional (PfMP)	Those who hold this credential are portfolio managers looking to prove their ability to manage and line up a portfolio of projects and programs to achieve organizational strategy and objectives.
PMI Professional in Business Analysis (PMI-PBA)	Designed for business analysts working on projects and programs, as well as project and program managers who perform business analysis as part of their role.
Program Management Professional (PgMP)	Candidates for this certification are usually already program managers, looking to validate their ability to manage complex, multiple projects and line up results to organizational goals. Professionals can use this certification to increase visibility and communicate valuable skills.

Systems Engineer Certification	Microsoft MCSE: Enterprise Devices and Apps	Designed for candidates with the skills needed to manage devices in today's bring-your-own-device (BYOD) enterprise. The candidate with this certification can be placed as a traditional desktop support technician to enterprise management of BYOD devices and apps.
	Microsoft MCSE: Messaging	For IT pros with an interest in cloud-based services such as Microsoft Office 365. This certification qualifies the candidate for a position in network and computer systems administration.
	Microsoft MCSE: Communication	Designed for those who have an interest in creating a consistent communications experience at the workplace. This credential qualifies candidates for a place in network and computer systems administration.
	Red Hat Certified Systems Administrator (RHCSA)	Designed for experienced system administrators looking to validate their skills and knowledge. It also can be helpful to students who have attended Red Hat System Administration I and II and want to be RHCSA certified.

	Red Hat Certified Engineer (RHCE)	For IT pros who are already RHCSAs and want to obtain higher-level credentials. Candidates such as experienced senior system administrators who have not yet been certified may be interested in this certification.
IT Security Certification	Global Information Assurance Certification (GIAC) Security Essentials (GSEC)	Designed for IT pros who want to demonstrate skills in IT systems hands-on roles with respect to security tasks. Ideal candidates for this certification possess an understanding of information security beyond simple terminology and concepts.
	International Information System Security Certification Consortium $(ISC)^2$ Certified Information Systems Security Professional (CISSP)	Ideal candidates for this certification are information assurance pros who know how to define the information system architecture, design, management, and controls to ensure the security of business environments.
	$(ISC)^2$ Systems Security Certified Practitioner (SSCP)	Designed for those with proven technical skills and practical security knowledge in hands-on operational IT roles. The SSCP provides confirmation of a practitioner's ability to implement, monitor, and administer IT infrastructure in accordance with information security policies and procedures that ensure data confidentiality, integrity, and availability.
	ISACA Certified Information Security Manager (CISM)	For candidates who have an inclination toward organizational security and want to demonstrate the ability to create a relationship between an information security program and broader business goals and objectives. This certification ensures knowledge of information security, development, and management of an information security program.

TABLE 17.5 Other Networking-Related Certification Programs

Certification Program	Description
Open Networking Foundation (ONF) Certified SDN Associate	The aim of this certification is to validate knowledge of foundational concepts in SDN.
ONF Certified SDN Engineer	Targeted at those SDN professionals who are actively engaged in the more technical components of the SDN ecosystem. In this case, the ONF-Certified SDN Engineer certification (OCSE) will validate the skills, knowledge, and abilities for technical professionals working in the SDN ecosystem.
HP ASE - SDN Application Developer	Certifies understanding of SDN environment and SDN application use cases and ability to write, test, and debug an SDN application.
VMware Certified Professional - Network Virtualization (VCP-NV)	Validates your ability to install, configure, and administer NSX virtual networking implementations, regardless of the underlying physical architecture.
Brocade NFV Certification	Designed for IT professionals to expand their skills and contributions to their company by gaining in-depth relevant NFV expertise.

TABLE 17.2 SDN Certification Programs

Certification Program	Description
Cisco Business Application Engineer Specialist	For application engineers who design, develop, and build business applications and who are looking to leverage the programmability capability of the new open network environment.
Cisco Network Programmability Developer Specialist	For software programmers who focus on the development of the network applications layer and will enable service provider, campus, and data center use cases. This certification and course develop the foundational skills needed to develop network applications in programmable environments.
Cisco Network Programmability Design Specialist	For engineers who have both architectural and application development expertise. You will learn how to better collect customer requirements and use this information combined with knowledge about the applications to leverage the infrastructure and translate requirements into a recommended open infrastructure.
Cisco Network Programmability Engineer Specialist	For engineers who deploy network applications into a programmable environment and make them operational. Key skills covered with this certification include implementing and troubleshooting an open network infrastructure designed by the network designers and architects.

TABLE 17.3 Network Virtualization Certification Programs

Certification Program	Description
Amazon Web Services (AWS) Certified Solutions Architect - Associate	Designed for IT professionals with experience in designing distributed applications and systems on the AWS platform.
AWS Certified Solutions Architect - Professional	Ideal candidates for this certification are professionals with advanced technical skills and experience in designing distributed applications and systems on the AWS platform.
AWS Certified Developer - Associate	Intended for those with technical skills and knowledge in developing and maintaining applications on AWS.
AWS Certified SysOps Administrator - Associate	Recognizes professionals with technical expertise in deployment, management, and operations on the AWS platform.
AWS Certified DevOps Engineer - Professional	Designed for those with expertise in provisioning, operating, and managing distributed application systems on AWS.
IBM Certified Solution Advisor - Cloud Computing Architecture V4	For IT pros who want to be recognized for their skills and experience in cloud computing. It aims to impart knowledge about cloud computing concepts and benefits, cloud computing design principles, IBM cloud computing architecture, and IBM cloud computing solutions.
IBM Certified Solution Architect - Cloud Computing Infrastructure V1	Recognizes those with proven knowledge in the design, planning, architecture, and management principles of an IBM cloud computing infrastructure.
Microsoft Certified Solution Expert: Private Cloud	For those who have interest towards Microsoft-led technologies and want to improve and prove their knowledge and skills. The certification provides the skills to build a private cloud solution with the help of Windows Server and System Center.

Microsoft Specialist Certification in Microsoft Azure	For IT pros such as developers who have prior experience working with Azure, Microsoft offers three specialist certifications to expand their skills with an eye toward future business requirements.
Salesforce Administrators	For those with experience being a Salesforce administrator.
Salesforce Implementation Experts	For those who have experience applying Sales Cloud solutions in a customer-facing role.
Salesforce Pardot Consultant	Designed for IT pros who have experience in applying Pardot marketing automation technology with in-depth knowledge of users and prospects, automation and segmentation tools, and building e-mails, forms, and landing pages in a customer-facing role.
Salesforce Developers	Intended for those who have cloud development experience and want to showcase their knowledge, skills, and abilities in creating bespoke application and analytic solutions using the Force.com platform.
Salesforce Technical Architect	For candidates who have experience measuring customer architecture and designing secure, high-performance technical solutions on the Force.com platform.
Google Qualified Developer	To be certified as a Google Qualified Developer, a candidate must pass at least one of the following exams: App Engine, Cloud Storage, Cloud SQL, BigQuery: Compute Engine.
Google Qualified Cloud Platform Developer	To be a certified Google Qualified Cloud Platform Developer, candidates must pass all five of the exams listed under Google Qualified Developer.

TABLE 17.4 Cloud Certification Programs

IT Skills

A global survey of 1156 respondents by TechPro Research [TECH14] revealed that many people fear that their current IT skill set will become obsolete. To stave off obsolescence, many respondents are planning to obtain additional IT certifications or degrees, with 57 percent planning for IT certifications either within their current job role or outside of their current job role. There are vast opportunities for the networking professional both to obtain these credentials and to leverage their education to maintain a high level of job security.

A useful tool in considering what specific skills you might need are the Dice rankings of skills in demand. Some of these are not directly related to networking tasks, but given the collaborative nature of the new networking environment, these skills can strengthen the network professional's resume. Table 17.6 shows the skills that commanded the highest salaries in the latest Dice salary survey, while Table 17.7 shows those skills for which demand is growing the fastest.

Skill	Description	Average Salary (US$)
PaaS	Platform as a Service	$130,081
Cassandra	A database management system developed by Facebook.	$128,646
MapReduce	A programming model from Google for processing huge data sets on large clusters of servers. It includes distribution, parallelizing, and fault-tolerant functions.	$127,315
Cloudera Impala	An open source MPP SQL query engine for mining data stored in Apache Hadoop clusters.	$126,816
Hbase	An open source, nonrelational, distributed database modeled after Google's BigTable and written in Java.	$126,369
Pig	A MapReduce programming tool used on Hadoop.	$124,563
ABAP	Advanced Business Application Programming. A high-level, COBOL-like programming language used to develop SAP applications.	$124,262
Chef	An open source configuration management tool.	$123,458
Flume	A service for collecting, aggregating, and moving large amounts of log data.	$123,816
Hadoop	An open source project from the Apache Software Foundation that provides a software framework for distributing applications on clusters of servers. Designed to handle huge amounts of data and inspired by Google's MapReduce programming model and file system.	$121,313

Source: 2015 Dice Tech Salary Survey

TABLE 17.6 Top-Paying Skills

Skill	Description	Average Salary (US$)
Cloudera Impala	An open source MPP SQL query engine for mining data stored in Apache Hadoop clusters.	$139,784
Adobe Experience Manager	Designed for organizing and managing creative assets, is popular among marketers, advertising-agency creative professionals and others who craft content.	$123,599
Ansible	System administrators rely on this open source tool to help them configure and manage PCs.	$124,860
Xamarin	Developers who want to rapidly build iOS and Android apps can use this tool to develop cross-platform in C#.	$101,707
OnCue	A web-based video streaming service.	$125,067
Laravel	An open source PHP Web application framework.	$96,219
RStudio	This integrated development environment for R (a statistical programming language that's proven a lucrative specialty for skilled developers) allows teams to share workspaces.	$117, 257
Unified Functional Testing	Gives tech pros the ability to comprehensively test software platforms and ecosystems.	$102,419
Pascal	Although Pascal has been around for 45 years, it is still very much in use.	$77,907
Apache Kafka	An open source tool developed by the Apache Software Foundation for maintaining real-time data feeds, capable of handling hundreds of megabytes of writes and reads per second from thousands of clients.	$134,950

Note: Descending order of popularity. These are not the skills most in demand, but the skills for which demand is growing most rapidly.

Source: Dice, April 2015

TABLE 17.7 Fastest Trending Skills

17.4 Online Resources

Numerous online resources can help you maintain and further your career, including the following:

- **ACM Career Resources:** ACM is an excellent source of CS career information. Resources include the following:
- **Online Resources for Graduating Students**, which has a useful list of links to career websites (http://www.acm.org/membership/membership/student/resources-for-grads).
- **ACM Career and Job Center (http://jobs.acm.org/)** is a place for job seekers and employers in the computing industry to connect with each other.
- **Computer Careers website (http://computingcareers.acm.org/)** provides guidance and resources for preparing for a career in computer science.

- **IEEE Resume Lab:** Online service that allows IEEE members to develop a resume or CV using specialized tools tailored for each step of the job seeking process. Excellent resource (https://ieee.optimalresume.com/index.php).
- **IEEE Computer Society Build Your Career (http://www.computer.org/web/careers):** Another excellent source of career information.
- **IEEE Job Site:** Yet another excellent source of career information, plus specific job leads (http://careers.ieee.org/).
- **ComputerWorld IT Topic Center (http://careers.ieee.org/):** Wide range of material, including news, white papers, career center, in-depth reports, and so on.
- **Computer Jobs (http://computerjobs.com/us/en/IT-Jobs/):** Lists thousands of searchable job opportunities categorized by major metropolitan markets and skill sets.
- **Career Overview (http://www.careeroverview.com/):** Contains jobs, job search websites and employment resources for professionals seeking career opportunities in computers, information technology, or another high-tech field. Good source of links.
- **DICE (http://www.dice.com/):** Frequently rated the best job site for positions worldwide in the information technology industry. Site also includes monthly article on timely topics, salary surveys, and discussions of skills in demand.

Another resource that you might find useful is the Computer Science Student Resources site that I maintain at http://www.computersciencestudent.com. This site is for professionals as well as students. The purpose of this site is to provide documents, information, and links for computer science students and professionals. Links and documents are organized into these categories:

Computer Science Student Resources

- **Math**: Includes a basic math refresher, a queuing analysis primer, a number system primer, and links to numerous math sites.
- **How-to**: Advice and guidance for literature searching, solving homework problems, writing technical reports, and preparing technical presentations.
- **Research resources**: Links to important collections of papers, technical reports, and bibliographies.
- **Writing**: A number of useful sites and documents for improving your writing skills.
- **Other useful**: A variety of other useful documents and links.
- **Careers**: Useful links and documents related to career building. This page includes links to all of the sites listed earlier in this chapter, plus more.

17.5 References

BORT13: Bort, J. “Will IT certs get you jobs and raises? Survey says yes.” *Network World*, November 14, 2011.

DICE13: Dice. “Why DevOps Is CPR for Cloud Applications.” *Dice Special Report*, November 2013.

DICE14: Dice. “Critical Skills for DevOps Engineers.” *Dice Special Report*, August 2014.

DICE15: Dice. “Spotlight on DevOps.” *Dice Special Report*, 2015.

HALE14: Hales, J. *SDN: How It Will Affect You and Why You Should Care.* Global Knowledge white paper, 2014.

MACV15: MacVitie, L. “Network Engineers: Don’t Fear the Code.” *Information Week*, March 2, 2015.

METZ14b: Metzler, J. *The Changing Role of the IT & Network Professional.* Webtorials, July 2014.

MICR15: Microsoft. *Enterprise DevOps.* Microsoft white paper, 2015.

MINI14: Minick, E., Rezabek, J., and Ring, C. *Application Release and Deployment for Dummies.* New York: Wiley, 2014.

PRET14: Pretz, K. “Five Skills for Managing Software-Defined Networks.” *IEEE The Institute*, December 2014.

SHAR15: Sharma, S., and Coyne. B. *DevOps for Dummies.* Hoboken, NJ: Wiley, 2015.

TECH14: TechPro Research. *The Future of IT Jobs: Critical Skills and Obsolescent Roles.* TechPro Research Report, August 2014.

Appendix A. References

In matters of this kind, everyone feels he is justified in writing and publishing the first thing that comes into his head when he picks up a pen, and thinks his own idea as axiomatic as the fact that two and two make four. If critics would go to the trouble of thinking about the subject for years on end and testing each conclusion against the actual history of war, as I have done, they would undoubtedly be more careful of what they wrote.

—*On War*, Carl von Clausewitz

Abbreviations

- **ACM**: Association for Computing Machinery
- **IEEE**: Institute of Electrical and Electronics Engineers
- **ITU-T**: International Telecommunication Union—Telecommunication Standardization Sector
- **NIST**: National Institute of Standards and Technology
- **RFC**: Request For Comments

References

AKAM15: Akamai Technologies. *Akamai's State of the Internet.* Akamai Report, Q4|2014. 2015.

BARI13: Bari, M. "PolicyCop: An Autonomic QoS Policy Enforcement Framework for Software Defined Networks," Proc. of IEEE SDN4FNS'13, Trento, Italy, Nov. 2013.

BENS11: Benson, T., et al. "CloudNaaS: A Cloud Networking Platform for Enterprise Applications." *Proceedings, SOCC'11*, October 2011.

BORT13: Bort, J. "Will IT certs get you jobs and raises? Survey says yes." *Network World*, November 14, 2011.

CISC14a: Cisco Systems. *Cisco Visual Networking Index: Forecast and Methodology, 2013–2018*. White Paper, 2014.

CISC14b: Cisco Systems. *The Internet of Things Reference Model*. White paper, 2014. http://www.iotwf.com/.

CISC14c: Cisco Systems. *Building the Internet of Things.* Presentation, 2014. http://www.iotwf.com/.

CISC15: Cisco Systems. *Internetworking Technology Handbook*. July 2015. http://docwiki.cisco.com/wiki/Internetworking_Technology_Handbook.

CISC15a: Cisco Systems. *Internetworking Technology Handbook*. July 2015.

http://docwiki.cisco.com/wiki/Internetworking_Technology_Handbook.

CISC15b: Cisco Systems. *Cisco IoT System: Deploy, Accelerate, Innovate.* Cisco white paper, 2015.

CLAR98: Clark, D., and Fang, W. "Explicit Allocation of Best-Effort Packet Delivery Service." *IEEE/ACM Transactions on Networking,* August 1998.

COGE13: Cogent Communications. *Network Services SLA Global.* October 2013. http://www.cogentco.com.

CSA11: Cloud Security Alliance. *Security as a Service (SecaaS).* CSA Report, 2011.

CSA13: Cloud Security Alliance. *The Notorious Nine Cloud Computing Top Threats in 2013.* CSA Report, February 2013.

DICE13: Dice. "Why DevOps Is CPR for Cloud Applications." *Dice Special Report*, November 2013.

DICE14: Dice. "Critical Skills for DevOps Engineers." *Dice Special Report*, August 2014.

DICE15: Dice. "Spotlight on DevOps." *Dice Special Report*, 2015.

ETSI14: ETSI TS 103 294 V1.1.1 Speech and Multimedia Transmission Quality (STQ); Quality of Experience; A Monitoring Architecture (2014-12).

FERG11: Ferguson, J., and Redish, A. "Wireless Communication with Implanted Medical Devices Using the Conductive Properties of the Body." *Expert Review of Medical Devices*, Vol. 6, No. 4, 2011. http://www.expert-reviews.com.

FOST13: Foster, N. "Languages for Software-Defined Networks." *IEEE Communications Magazine*, February 2013.

FRAH15: Frahim, J., et al. *Securing the Internet of Things: A Proposed Framework.* Cisco white paper, March 2015.

GUPT14: Gupta, D., and Jahan, R. *Securing the Internet of Things: A Proposed Framework.* Tata Consultancy Services White Paper, 2014. http://www.tcs.com.

HALE14: Hales, J. *SDN: How It Will Affect You and Why You Should Care.* Global Knowledge white paper, 2014.

HAWI14: Hawilo, H., et al. "NFV: State of the Art, Challenges, and Implementation in Next Generation Mobile Networks." *IEEE Network*, November/December 2014.

HOGG14: Hogg, S. "SDN Security Attack Vectors and SDN Hardening." *Network World*, Oct 28, 2014.

HOSS13: Hossfeld, T., et al. "Internet Video Delivery in YouTube: From Traffic Measurements to Quality of Experience." Book chapter in *Data Traffic Monitoring and Analysis: From Measurement, Classification, and Anomaly Detection to Quality of Experience*, Lecture Notes in Computer Science, Volume 7754, 2013.

IBM11: IBM Study, "Every Day We Create 2.5 Quintillion Bytes of Data." Storage Newsletter, October 21, 2011. http://www.storagenewsletter.com/rubriques/market-

reportsresearch/ibm-cmo-study/.

ISGN12: ISG NFV. *Network Functions Virtualization: An Introduction, Benefits, Enablers, Challenges & Call for Action*. ISG NFV white paper, October 2012.

ITUT12: ITU-T. Focus Group on Cloud Computing Technical Report Part 3: Requirements and Framework Architecture of Cloud Infrastructure. FG Cloud TR, February 2012.

KAND12: Kandula, A., Sengupta, S., and Patel, P. "The Nature of Data Center Traffic: Measurements and Analysis." ACM SIGCOMM Internet Measurement Conference, November 2009.

KETY10: Ketyko, I., De Moor, K., Joseph, W., and Martens, L. "Performing QoE-Measurements in an Actual 3G Network," IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, March 2010.

KHAN09: Khan, A., Sun, L., and Ifeachor, E. "Content Clustering Based Video Quality Prediction Model for MPEG4 Video Streaming over Wireless Networks," *IEEE International Conference on Communications*, 2009.

KHAN15: Khan, F. *A Beginner's Guide to NFV Management & Orchestration (MANO)*. Telecom Lighthouse. April 9, 2015. http://www.telecomlighthouse.com.

KIM14: Kim, H., and Choi, S. "QoE Assessment Model for Multimedia Streaming Services Using QoS Parameters," *Multimedia Tools and Applications*, October 2014.

KRAK09: Krakowiak, S. *Middleware Architecture with Patterns and Frameworks.* 2009. http://sardes.inrialpes.fr/%7Ekrakowia/MW-Book/.

KREU15: Kreutz, D., et al. "Software-Defined Networking: A Comprehensive Survey." *Proceedings of the IEEE*, January 2015.

KUIP10: Kuipers, F. et al. "Techniques for Measuring Quality of Experience," 8th International Conference on Wired/Wireless Internet Communications, 2010.

KUMA13: Kumar, R. Software Defined Networking—a Definitive Guide. Smashwords.com, 2013.

MA14: Ma, H., Seo, B., and Zimmermann, R. "Dynamic Scheduling on Video Transcoding for MPEG DASH in the Cloud Environment," Proceedings of the 5th ACM Multimedia Systems Conference, March 2014.

MACV15: MacVitie, L. "Network Engineers: Don't Fear the Code." *Information Week*, March 2, 2015.

MARS06: Marsh, I., Grönvall, B., and Hammer, F. "The Design and Implementation of a Quality-Based Handover Trigger," 5th International IFIP-TC6 Networking Conference, Coimbra, Portugal.

MCEW13: McEwen, A., and Cassimally, H. *Designing the Internet of Things.* New York: Wiley, 2013.

MCMU14: McMullin, M. "SDN is from Mars, NFV is from Venus." *Kemp Technologies Blog*, November 20, 2014. http://kemptechnologies.com/blog/sdn-mars-nfv-venus.

METZ14a: Metzler, J. *The 2015 Guide to SDN and NFV.* Webtorials, December 2014.

METZ14b: Metzler, J. *The Changing Role of the IT & Network Professional.* Webtorials, July 2014.

MICR15: Microsoft. *Enterprise DevOps.* Microsoft white paper, 2015.

MINI14: Minick, E., Rezabek, J., and Ring, C. *Application Release and Deployment for Dummies.* New York: Wiley, 2014.

MOLL12: Moller, S., Callet, P., and Perkis, A. "Qualinet White Paper on Definitions on Quality of Experienced," European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003) (2012).

MURP07: Murphy, L. et al. "An Application-Quality-Based Mobility Management Scheme," Proceedings of 9th IFIP/IEEE International Conference on Mobile and Wireless Communications Networks, 2007.

NAKI15: Nakina Systems. *Achieving Security Integrity in Service Provider NFV Environments.* Nakina Systems white paper, 2015.

NETW14: Network World. Survival Tips for Big Data's Impact on Network Performance. White paper. April 2014.

NGUY13: Nguyen, X., et al. "Efficient Caching in Content-Centric Networks using OpenFlow," 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2013.

NGUY14: Nguyen, X., Saucez, D,, and Thierry, T. "Providing CCN Functionalities over OpenFlow Switches," hal-00920554, 2013. https://hal.inria.fr/hal-00920554/.

ODCA14: Open Data Center Alliance. Open Data Center Alliance Master Usage Model: Software-Defined Networking Rev. 2.0. White Paper. 2014.

ONF12: Open Networking Foundation. *Software-Defined Networking: The New Norm for Networks.* ONF White Paper, April 13, 2012.

ONF14: Open Networking Foundation. *OpenFlow-Enabled SDN and Network Functions Virtualization.* ONF white paper, February 17, 2014.

POTT14: Pott, T. "SDI Wars: WTF Is Software Defined Center Infrastructure?" *The Register*, October 17, 2014. http://www.theregister.co.uk/2014/10/17/sdi_wars_what_is_software_defined_infrastruct

PRET14: Pretz, K. "Five Skills for Managing Software-Defined Networks." *IEEE The Institute*, December 2014.

QUIN12: M.R.Quintero, M., and Raake, A. "Is Taking into Account the Subjects' Degree of Knowledge and Expertise Enough When Rating Quality?" Fourth International Workshop on Quality of Multimedia Experience (QoMEX), pp.194,199, 5–7 July 2012.

SCHE13: Scherz, P., and Monk, S. *Practical Electronics for Inventors.* New York: McGraw-Hill, 2013.

SCHN14: Schneier, B. “The Internet of Things is Wildly Insecure—and Often Unpatchable.” *Wired*, January 6, 2014.

SDNC14: SDNCentral. SDNCentral Network Virtualization Report, 2014 Edition, 2014.

SEGH12: Seghal, A., et al. “Management of Resource Constrained Devices in the Internet of Things.” *IEEE Communications Magazine*, December 2012.

SHAR15: Sharma, S., and Coyne. B. *DevOps for Dummies*. Hoboken, NJ: Wiley, 2015.

SHEN11: Schenker, S. “The Future of Networking, and the Past of Protocols,” October 2011.Video: http://www.youtube.com/watch?v=YHeyuD89n1Y; Slides: http://www.slideshare.net/martin_casado/sdn-abstractions.

STAL15a: Stallings, W., and Brown, L. *Computer Security: Principles and Practice.* Englewood Cliffs, NJ: Pearson, 2015.

STAL15b: Stallings, W. *Cryptography and Network Security*. Englewood Cliffs, NJ: Pearson, 2015.

STAN14: Stankovic, J. “Research Directions for the Internet of Things.” *Internet of Things Journal*, Vol. 1, No. 1, 2014.

SZIG14: Szigeti, T., Hattingh, C., Barton, R., and Briley, K. *End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks.* Englewood Cliffs, NJ: Pearson. 2014.

TECH14: TechPro Research. *The Future of IT Jobs: Critical Skills and Obsolescent Roles.* TechPro Research Report, August 2014.

VAQU14: Vaquero, L., and Rodero-Merino, L. “Finding Your Way in the Fog: Towards a Comprehensive Definition of Fog Computing.” *ACM SIGCOMM Computer Communication Review*, October 2014.

WANG12: Wang, G.; Ng, E.; and Shikh, A. “Programming Your Network at Run-Time for Big Data Applications.” *Proceedings, HotSDN’12*. August 13, 2012.

XI11: Xi, H. “Bandwidth Needs in Core and Aggregation Nodes in the Optical Transport Network.” IEEE 802.3 Industry Connections Ethernet Bandwidth Assessment Meeting, November 8, 2011. http://www.ieee802.org/3/ad_hoc/bwa/public/nov11/index_1108.html.

Glossary

In studying the Imperium, Arrakis, and the whole culture which produced Maud'Dib, many unfamiliar terms occur. To increase understanding is a laudable goal, hence the definitions and explanations given below.

—*Dune*, Frank Herbert

3G: Third-generation wireless cellular communications technology. Designed to provide fairly high-speed wireless communications to support multimedia, data, and video in addition to voice. Target data rates are 144 and 384 kbps. Some 3G systems also provide support up to 2 Mbps for office use.

4G: Fourth generation wireless cellular communications technology. Based on all-IP packet switched network. Support peak data rates of up to approximately 100 Mbps for high-mobility mobile access and up to approximately 1 Gbps for low-mobility access such as local wireless access.

5G: Projected fifth-generation wireless cellular communications technology. The focus for 5G will be on building more intelligence into the network, to meet service quality demands by dynamic use of priorities, adaptive network reconfiguration, and other network management techniques.

access network: A network that connects directly to the end user or customer.

accuracy: The closeness of agreement between the result of a measurement and the true value of the measure. It can be expressed as a qualitative assessment of correctness, or freedom from error, or a quantitative measure of the expected magnitude of error.

actuator: A device that accepts an electrical signal and converts it into a physical, chemical, or biological action.

analytics: Analysis of massive amounts of data, particularly with a focus on decision making.

application lifecycle management: The administration and control of an application from inception to its demise. It embraces requirements management, system design, software development, and configuration management, and it implies an integrated set of tools for developing and controlling the project.

application programming interface (API): A language and message format used by an application program to communicate with the operating system or some other control program such as a database management system (DBMS) or communications protocol. APIs are implemented by writing function calls in the program, which provide the linkage to the required subroutine for execution. An open or standardized API can ensure the portability of the application code and the vendor independence of the called service.

application provider: An entity generating/selling user applications to be executed on the user's platform.

application service provider: An organization that hosts software applications within its own facilities. It provides network-accessible applications such as e-mail, web hosting, banking, and cloud-based services.

attack surface: The reachable and exploitable vulnerabilities in a system.

attack vector: The method or type of attack on a computer system or network.

autonomous system (AS): A network that is administered by a single set of management rules that are controlled by one person, group or organization. Autonomous systems often use only one routing protocol, although multiple protocols can be used. The core of the Internet is made up of many autonomous systems.

backbone network: Same as core network.

best effort: A network or Internet delivery technique that does not guarantee delivery of data and treats all packets equally. All packets are forwarded on a first-come, first-served basis. Preferential treatment based on priority or other concerns is not provided.

big data: A collection of data on such a large scale that standard data analysis and management tools are not adequate. More broadly, big data refers to the volume, variety and velocity of structured and unstructured data pouring through networks into processors and storage devices, along with the conversion of such data into business advice for enterprises.

blade server: A server architecture that houses multiple server modules (blades) in a single chassis. It is widely used in data centers to save space and improve system management. Either self-standing or rack mounted, the chassis provides the power supply, and each blade has its own CPU, memory, and hard disk.

broadcast: An address recognized by all hosts on a network or within a network domain. With broadcast addressing, one transmission stream is used to each switch, at which point data are distributed out to the end users on separate lines.

business support system (BSS): Software applications that support customer-facing activities. Billing, order management, customer relationship management and call center automation are all BSS applications. BSS may also encompass the customer-facing veneer of OSS application such as trouble-ticketing and service assurance; these are back-office activities but initiated directly by contact with the customer.

capital expenditure (CapEx): A business expense incurred to create future benefits. A CapEx is incurred when a business spends money either to buy fixed assets or to add to the value of an existing asset with a useful life that extends beyond the tax year.

client/server: A common form of distributed system in which software is split between server tasks and client tasks. A client sends requests to a server, according to some protocol, asking for information or action, and the server responds.

cloud computing: A loosely defined term for any system providing access via the Internet to processing power, storage, software or other computing services, often via a web browser. Often, these services are rented from an external company that hosts and manages them.

commercial off-the-shelf (COTS): Item that is commercially available, leased, licensed, or sold to the general public and that requires no special modification or maintenance over the lifecycle of the product to meet the needs of the procuring agency.

Communication as a Service (CaaS): A service offered via cloud computing in which the capability provided to the cloud service customer is real time interaction and collaboration.

congestion: The condition of a network when there is not enough capacity to support the current traffic load.

congestion control: Protocol mechanisms for relieving or avoiding congestion.

consortium: A group of independent organizations joined by common interests. In the area of standards development, a consortium typically consists of individual corporations and trade groups concerned with a specific area of technology.

constrained device: In an IoT, a device with limited volatile and nonvolatile memory, limited processing power, and a low-data-rate transceiver.

container: Hardware or software that provides an execution environment for software.

container virtualization: A technique where the underlying operating environment of an application is virtualized. This will commonly be the operating system kernel, and the result is an isolated container in which the application can run.

content provider: An organization or individual that creates information, including educational or entertainment content distributed via the Internet or enterprise networks. A content provider may or may not provide the software used to access the material.

core network: A central network that provides networking services to attached distribution and access networks. Also referred to as a backbone network.

core router: A router that resides within the middle of the network rather than at its periphery. The routers that make up the backbone of the Internet are core routers.

cross-section bandwidth: For a network, this is the maximum bidirectional data rate that can pass between two parts of the network if it is divided into two equal halves. Also referred to as *bisection bandwidth*.

data deduplication: The elimination of redundant data. It includes (1) compressing data by only storing changes to data, and (2) replacing duplicate copies of chunks of data or files with pointers to a single copy.

datagram: A packet that is treated independently of other packets for packet switching. A datagram carries information sufficient for routing from the source to the destination

without the necessity of establishing a logical connection between the endpoints.

deep packet inspection: Analyzing network traffic to discover the type of application that sent the data. In order to prioritize traffic or filter out unwanted data, deep packet inspection can differentiate data, such as video, audio, chat, Voice over IP (VoIP), e-mail, and web. Inspecting the packets all the way up to the application layer, it can be used to analyze anything within the packet that is not encrypted. For example, it can determine not only that the packets contain the contents of a web page but also which website the page is from.

delay jitter: The variation in delay associated with the transfer of packets between two points. Typically measured as the maximum variation in delay experienced by packets in a single session.

denial of service (DoS): The prevention of authorized access to resources or the delaying of time-critical operations.

DevOps (development operations): The tighter integration between the developers of applications and the IT department that tests and deploys them. DevOps is said to be the intersection of software engineering, quality assurance, and operations.

differentiated services: Functionality in the Internet and private internets to support specific QoS requirements for a group of users, all of whom use the same service label in IP packets.

differentiated services codepoint (DSCP): A 6-bit field in the IP header that is used to classify packets for differentiated services (a form of QoS traffic management).

distributed denial of service (DDoS): An attack when multiple systems are used to flood servers or network devices or links with traffic in an attempt to overwhelm its available resources (bandwidth, memory, processing power, and so on), making it unavailable to respond to legitimate users.

distribution network: Connects access networks to a core network.

edge router: A router that sits at the periphery of a network. Also called an *access router* or *aggregation router*.

elastic traffic: Network traffic that is tolerant to variations in delay, jitter, and throughput. Typically carried over TCP or UDP.

embedded system: Any device that includes a computer chip, but that is not a general-purpose workstation, desktop or laptop computer.

electronic product code (EPC): A standard code for RFID tags. The EPC ranges from 64 to 256 bits and contains, at minimum, the product number, serial number, company ID and EPC version. Several bodies are involved in developing standards, including GS1 and EPCglobal.

end user: The ultimate consumer of applications, data and services on a computing platform.

Ethernet: The commercial name for a wired local-area network technology. It involves the use of a shared physical medium, a medium access control protocol, and transmission of data in packets. Standards for Ethernet products are defined by the IEEE 802.3 committee.

exterior router protocol (ERP): A protocol that distributes routing information to collaborating routers that connect autonomous systems. BGP is an example of an ERP. Historically, referred to as an exterior gateway protocol.

flow: A sequence of packets between a source and destination that are recognized by the network as related and are treated in a uniform fashion.

fog computing: A scenario in which a massive number of heterogeneous, decentralized devices communicate with each other and with the network to perform storage and processing tasks without the intervention of third parties.

hardware virtualization: The use of software to partition a computer's resources into separate and isolated entities called virtual machines. It enables multiple copies of the same or different operating systems to execute on the computer and prevents applications from different virtual machines from interfering with each other.

high-availability (HA) cluster: A multiple-computer architecture consisting of redundant network nodes that deliver a secondary or backup service when the primary service fails. Such clusters build redundancy into their computing environments to eliminate single points of failure, and they can incorporate multiple network connections, redundant data storage volumes, doubled-up power supplies, and other backup components and capabilities.

hypervisor introspection: The hypervisor capability to monitor each guest OS or virtual machine as it is running, for security purposes.

IEEE 802: A committee of the Institute of Electrical and Electronics Engineers (IEEE) responsible for developing standards for local- and metropolitan-area networks (LANs and MANs).

IEEE 802.1: An IEEE 802 working group responsible for developing standards in the following areas: 802 LAN/MAN architecture, internetworking among 802 LANs, MANs and other wide-area networks, 802 Security, 802 overall network management.

IEEE 802.3: An IEEE 802 working group responsible for developing standards for Ethernet local-area networks (LANs).

inelastic traffic: Network traffic that is relatively intolerant to variations in delay, jitter, and throughput. Real-time traffic is an example of inelastic traffic.

information technology (IT): The common term for the entire spectrum of technologies for information processing, including software, hardware, communications technologies, and related services. In general, IT does not include embedded technologies that do not generate data for enterprise use.

Infrastructure as a Service (IaaS): A group of capabilities offered via cloud computing in

which the cloud service customer can provision and use processing, storage, or networking resources.

interior router protocol (IRP): A protocol that distributes routing information to collaborating routers within an autonomous system. RIP and OSPF are examples of IRPs. Historically, referred to as an interior gateway protocol.

Internet: A worldwide internetwork based on TCP/IP that interconnects thousands of public and private networks and millions of users.

internet (with lower case "i"): A large network made up of a number of smaller networks. Also referred to as an *internetwork*.

Internet of Things (IoT): The expanding connectivity, particularly via the Internet of a wide range of sensors, actuators, and other embedded systems. In almost all cases, there is no human user, with interaction fully automated.

Internet Protocol (IP): A standardized protocol that executes in hosts and routers to interconnect a number of independent networks.

IP security (IPsec): Suite of protocols for securing IP communications at the network layer by authenticating and/or encrypting each IP packet in a data stream. IPsec also includes protocols for cryptographic key management.

LAN switch: A packet-forwarding network device for (1) interconnecting end systems in a local area to form a local-area network (LAN) segment, (2) connecting with other LAN switches to for a larger LAN, and (3) providing connection to routers and other network devices for wide-area network connectivity.

Layer 3 (L3) switch: A high-performance device for network routing. Layer 3 switches are very similar to routers. The key difference between L3 switches and routers is that a L3 switch replaces some of a router's software logic with hardware to offer better performance. L3 switches often cost less than traditional routers. Designed for use within local networks, a Layer 3 switch will typically not possess the WAN ports and wide-area network features a traditional router has.

media access control (MAC) frame: : A group of bits that includes source and destination addresses and other protocol control information plus, optionally, data. It is the basic unit of transmission on Ethernet and Wi-Fi LANs.

microcontroller: A single chip that contains the processor, non-volatile memory for the program (ROM or flash), volatile memory for input and output (RAM), a clock and an I/O control unit. Also called a *computer on a chip*.

microprocessor: A processor whose elements have been miniaturized into one or a few integrated circuits.

Multiprotocol Label Switching (MPLS): A protocol developed the IETF for directing packets in a wide-area IP network, or other WAN. MPLS adds a 32-bit label to each packet to improve network efficiency and to enable routers to direct packets along predefined

routes in accordance with the required quality of service.

Network as a Service (NaaS): A service offered via cloud computing in which the capability provided to the cloud service customer is transport connectivity and related network capabilities.

network convergence: The provision of telephone, video, and data communication services within a single network.

network interface card: An adapter circuit board installed in a computer to provide a physical connection to a network.

network functions virtualization: The virtualization of network functions by implementing these functions in software and running them on virtual machines.

network operating system (NOS): A server-based operating system oriented to computer networking. It may include directory services, network management, network monitoring, network policies, user group management, network security and other network-related functions.

network provider: An organization that delivers communications services over a typically large geographic area. It provides, maintains, and manages network equipment and networks, either public or private.

northbound API: In an SDN environment, the interface between the control and application planes.

Open Service Gateway Initiative (OSGi): A set of specifications that defines a dynamic component system for Java. These specifications reduce software complexity by providing a modular architecture for large-scale distributed systems as well as small, embedded applications.

open standard: A standard that is developed on the basis of an open decision-making procedure available for implementation to all interested parties, that is available to all on a royalty-free basis, and that is intended to promote interoperability among products from multiple vendors.

operational expenditure (OpEx): Refers to business expenses incurred in the course of ordinary business, such as maintenance and operation of equipment.

operational technology (OT): Hardware and software that detects or causes a change through the direct monitoring and/or control of physical devices, processes and events in the enterprise.

operations support system (OSS): Software (occasionally hardware) applications that support back-office activities which operate a network, and provision and maintain customer services. OSS is typically used by network planners, service designers, operations, architects, support, and engineering teams in the service provider.

packet: A unit of data sent across a network. A packet is a group of bits that includes data

plus protocol control information. The term generally applies to protocol data units at the network layer.

packet forwarding: The function performed by a router of accepting a packing on an input link and transmitting it on an output link.

packet switching: A method of transmitting messages through a communications network, in which long messages are subdivided into short packets. Each packet is passed from source to destination through intermediate nodes. At each node, the entire message is received, stored briefly, and then forwarded to the next node.

peer: On the same level or providing the same function. In networking, a peer is a node that provides the same functionality as another. For example, two desktop PCs in a network are peers. A desktop PC and a server are not peers as they perform different operations. The desktop PC may query the server for business data, but the server does not query the PC for the same data.

peering: An agreement between two routers to accept each other's data packets and forward them. A peer relationship generally involves the exchange of routing information.

Platform as a Service (PaaS): A group of capabilities offered via cloud computing in which the cloud service customer can deploy, manage and run customer-created or customer-acquired applications using one or more programming languages and one or more execution environments supported by the cloud service provider.

Power over Ethernet (PoE): Distributing power over an Ethernet cable to a target device that is not plugged into an AC wall outlet. PoE enables remote network devices in locations far away from AC sources.

powerline carrier (PLC): A data network that uses a building's electrical system as the transmission medium and regular wall outlets as connecting points. It is commonly used to extend a wired Ethernet network into another room.

precision: The degree of agreement of repeated measurements of the same property, expressed quantitatively as the standard deviation computed from the results of the series of measurements.

protocol: A set of semantic and syntactic rules that describe how to transmit data, especially across a network. Low-level protocols define the electrical and physical standards to be observed, bit- and byte-ordering, and the transmission and error detection and correction of the bit stream. High-level protocols deal with the data formatting, including the syntax of messages, semantics of messages, character sets, and sequencing of messages.

protocol architecture: The software structure that implements the communications function. Typically, the protocol architecture consists of a layered set of protocols, with one or more protocols at each layer.

protocol control information: Information exchanged between entities of a given layer, via the service provided by the next lower layer, to coordinate their joint operation.

protocol data unit (PDU): Information that is delivered as a unit between peer entities of a network. A PDU typically contains control information and address information in a header. The PDU may also contain data.

quality of experience (QoE): A subjective measure of performance in a system. QoE relies on human opinion and differs from quality of service (QoS), which can be precisely measured.

quality of service (QoS): The measurable end-to-end performance properties of a network service, which can be guaranteed in advance by a service level agreement between a user and a service provider, so as to satisfy specific customer application requirements. Note: These properties may include throughput (bandwidth), transit delay (latency), error rates, priority, security, packet loss, packet jitter, and so on.

radio-frequency identification (RFID): A data collection technology that uses electronic tags attached to items to allow the items to be identified and tracked by a remote system. The tag consists of an RFID chip attached to an antenna.

real time: As fast as required. A real-time system must respond to a signal, event or request fast enough to satisfy some requirement.

real-time traffic: A data flow that must meet real-time requirements, such as low jitter and low delay.

Request For Comments (RFC): A document in the archival series that is the official channel for publications of the Internet Society, including IETF and IRTF publications. An RFC may be informational, best practice, draft standard, or an official Internet Standard.

resolution: The smallest distinguishable increment into which a measured quantity is divided.

role-based access control (RBAC): Controls access based on the roles that users have within the system and on rules stating what accesses are allowed to users in given roles.

router: A network device that forwards data packets from one network to another. The forwarding decision is based on network layer information and routing tables, often constructed by routing protocols. Routers require packets formatted in a routable protocol, the global standard being the Internet Protocol (IP).

routing: The determination of a path that a data unit (frame, packet, message) will traverse from source to destination.

routing protocol: A protocol used by routers to determine the appropriate path onto which data should be forwarded. The routing protocol also specifies how routers report changes and share information with the other routers in the network that they can reach.

scale out: Expand the capability of a single physical machine or virtual machine.

scale up: Expand capability by adding additional physical or virtual machines.

sensor: A device that converts a physical, biological, or chemical parameter into an

electrical signal.

service provider: A network-accessible entity that can provide services to an end user.

Software as a Service (SaaS): A group of capabilities offered via cloud computing in which the cloud service customer can use the cloud service provider's applications.

software-defined networking (SDN): An approach to designing, building and operating large-scale networks based on programming the forwarding decisions in routers and switches via software from a central server. SDN differs from traditional networking, which requires configuring each device separately and which relies on protocols that cannot be altered.

software-defined storage (SDS): An approach to data storage management and use in which the software that controls storage-related tasks is decoupled from the physical storage hardware.

southbound API: In an SDN environment, the interface between the control and data planes.

standard: A document that provides requirements, specifications, guidelines, or characteristics that can be used consistently to ensure that materials, products, processes, and services are fit for their purpose. Standards are established by consensus among those participating in a standards-making organization and are approved by a generally recognized body.

standards-developing organization (SDO): An official national, regional, or international standards body that develop standards and/or that coordinate the standards activities of a specific country, region or the world. Some SDOs facilitate the development of standards through support of technical committee activities, and some may be directly involved in standards development.

TCP/IP protocol architecture: The protocol architecture built around the TCP and IP protocols, consisting of five layers: physical, data link, network/internet (usually IP), transport (usually TCP or UDP), and application.

token bucket: A data-flow control mechanism that adds tokens in periodical time intervals into a buffer (bucket) and allows a data packet to leave the sender only if there are at least as many tokens in the bucket as the packet length of the data packet. This strategy allows precise control of the time interval between two data packets in the network.

top-of-rack (ToR) switch: A blade server arrangement in which servers connect to one or two Ethernet switches installed inside the rack. The actual physical location of the switch does not necessarily need to be at the top of the rack. Other switch locations could be bottom of the rack or middle of rack. However, top of the rack is most common due to easier accessibility and cleaner cable management.

traffic engineering: That aspect of network engineering dealing with the issues of performance evaluation and performance optimization of operational networks. Traffic engineering encompasses the application of technology and scientific principles to the

measurement, characterization, modeling, and control of network traffic.

transceiver: A device that can both transmit and receive information.

unicast: An address which only one host will recognize. With unicast addressing, even though multiple users might request the same data from the same server at the same time, duplicate data streams are transmitted, one to each user.

unified communications: The integration of real-time, enterprise, communication services such as instant messaging, presence information, voice (including IP telephony), web and video conferencing, and speech recognition with non-real-time communication services such as unified messaging (integrated voicemail, e-mail, SMS, and fax).

Uniform Resource Identifier (URI): A compact sequence of characters that identifies an abstract or physical resource. The URI specification (RFC 3986) defines a syntax for encoding arbitrary naming or addressing schemes, and provides a list of such schemes. The URL (Uniform Resource Locator) is a type of URI, in which an access protocol is designated and a specific Internet address is provided.

virtual local-area network (VLAN): A virtual network abstraction on top of a physical packet-switched network. A VLAN is essentially a broadcast domain for a specified set of switches. These switches are required to be aware of the existence of VLANs and configured accordingly, to perform switching of packets between devices belonging to the same VLAN.

virtual machine: One instance of an operating system along with one or more applications running in an isolated partition within the computer. It enables different operating systems to run in the same computer at the same time as well as prevents applications from interfering with each other.

virtual machine monitor (VMM): A system program that provides a virtual machine environment. Also called a *hypervisor*.

virtual network: An abstraction of physical network resources as seen by some upper software layer. Virtual network technology enables a network provider to support multiple virtual networks that are isolated from one another. Users of a single virtual network are not aware of the details of the underlying physical network or of the other virtual network traffic sharing the physical network resources.

virtual private network (VPN): The use of encryption and authentication in the lower protocol layers to provide a secure connection through an otherwise unsecure network, typically the Internet. VPNs are generally cheaper than real private networks using private lines but rely on having the same encryption and authentication system at both ends. The encryption may be performed by firewall software or possibly by routers.

virtualization: A variety of technologies for managing computer resources by providing an abstraction layer between the software and the physical hardware. These technologies effectively emulate or simulate a hardware platform, such as a server, storage device, or network resource, in software.

Wi-Fi: Refers to the wireless LAN technology standardized by the IEEE 802.11 committee. The term Wi-Fi designates products that have been certified by the Wi-Fi Alliance to conform to the 802.11 standards and have passed interoperability tests.

Index

Symbols

A

B

C

D

E

F

G

H

I

J–K

L

M

N

O

P

Q

R

S

T

U

V

W

X – Z

Code Snippets

```
def switch_join(s):
    pat1 = {inport:1}
    pat2web = {inport:2, srcport:80}
    pat2 = {inport:2}
    install(s, pat1, DEFAULT, [fwd(2)])
    install(s, pat2web, HIGH, [fwd(1)])
    install(s, pat2, DEFAULT, [fwd(1)])
    query_stats(s, pat2web)
def stats_in(s, xid, pat, pkts, bytes):
    print bytes
    sleep(30)
    query_stats(s, pat)
```

```
def repeater():
    rules=[Rule(inport:1, [fwd(2)])
           Rule(inport:2, [fwd(1)])]
    register(rules)
def web monitor():
    q = (Select(bytes) *
       Where(inport=2 & srcport=80) *
       Every(30))
    q >> Print()
def main():
    repeater()
    monitor()
```