

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

chapter

6

Architecture and Infrastructure

This chapter provides managers with an overview of IT architecture and infrastructure issues and designs. It begins by translating a business into IT architecture and then from the architecture into infrastructure. The manager's role is then discussed, and an example of a fictitious company, GiantCo.com, is used to show how strategy leads to infrastructure. The framework used to describe the basic components of architecture and infrastructure, introduced in Chapter 1, is revisited here, providing a language and structure for describing hardware, software, network, and data considerations. Common architectures are then presented, including centralized, decentralized and Web-based service-oriented architecture (SOA). Architectural principles are covered, followed by a discussion of enterprise architecture. Virtualization and cloud computing, two current architectural considerations, are reviewed. The chapter concludes with a discussion of managerial considerations that apply to any architecture.

Mohawk,¹ a paper mill in upstate New York, was established in 1931. Contrary to a common assumption that information technology is not critical to old technology industry players facing a declining market, the firm has not only embraced cloud computing but also has been able to transform its business because of the cloud in three ways: (1) moving from manufacturing as its primary focus to providing service, (2) shifting from a self-sufficient model to one of collaboration with a network of partners, and (3) ensuring that the partner network is flexible and its capabilities are integrated with those of Mohawk. Mohawk accomplished this flexibility by using service-oriented architecture (SOA) tools, which enable a firm to scale technology services (and expenses) up and down instantaneously according to its needs.² Also, applications under SOA can be added or subtracted as needed.

Mohawk's new envelope manufacturing facility serves as a vivid example to illustrate the benefits of flexibility. Along the way, the company learned of the anticipated bankruptcy of the largest envelope manufacturing firm in the United States and developed a list of six outsourced firms to turn its premium papers into envelopes. After six months of using those suppliers and investing in building its own in-house envelope manufacturing capabilities, Mohawk was able to shift to an insourcing model for 90% of its volume. The cloud services approach avoided the information systems difficulties usually inherent in such a transformation.

There are also benefits to internal flexibility as well. As processing volumes increase and decrease, sometimes on a seasonal basis and sometimes due to new or discontinued lines of business, Mohawk experiences corresponding increases and decreases in its requirements for space, servers, and processing. Its cloud approach allows the company to set up or dismantle servers quickly.

¹ Adapted from Paul J. Stumas, Michelle L. Kaurst-Brown, and Scott A. Bernard, "The Business Transformation Payoffs of Cloud Services at Mohawk," *MIS Quarterly Executive* 13, no. 4 (2014).

² Christopher Hale, "Liaison Technologies to Deliver SOA-in-the-Cloud Services to Mohawk Papers," *Business Wire* (February 24, 2010), http://www.businesswire.com/news/home/20100224006065/en/Liaison-Technologies-Deliver-SOA-in-the-Cloud-Services-Mohawk-Papers#.VYFh_0ZZWjs (accessed June 17, 2015).

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Mohawk's experience shows that cloud computing is not just a mechanism to avoid or reduce costs or to gain operational benefits. The cloud can enable transformation of the business itself. Mohawk's mission changed from "making paper" to "making connections," which involves being able to sell directly to consumers five times the number of products than in the pre-2011 period when it mainly sold a few lines of paper to 10–15 large distributors. Partners now offer many of those products, and the system provides the capabilities to sell from Mohawk's own inventory or from the partners in a seamless way directly to many thousands of small businesses and consumers via its Web site.

Mohawk was able to make the changes it believed were necessary by shifting from an electronic data interchange (EDI) approach to a simpler, more interchangeable format using XML and other tools. Liaison Technologies, its integration consulting firm, enabled these changes by first developing what it calls a cloud integration platform and building upon that platform in several stages to ultimately arrive at an enhanced Web services platform that enabled other organizations and customers to request information, inquire about freight charges and pricing, place orders, and pay for their orders through connections with banks. The platform enables designers to "mash up" (combine) applications as needed on Web sites that can be built rather quickly. Each feature "plugs in" using tools that make it easy to connect the Web sites to existing databases.

Payoffs to Mohawk included:

- Shaking the precloud annual earnings decreases of 2%–5% per year to tripling its earnings in two years
- Automating its transaction processes, saving \$1 million to \$2 million annually in staff costs
- Increasing its product variety fivefold
- Increasing its customer base from 10–15 distributors to 100 business partners and many thousands of direct customers

Not all firms can base their entire operations on a cloud platform that permits integration with other organizations. Mohawk's experiences can be considered to be "cutting edge," and integration consulting is a rather new phenomenon. Further, even if firms use a cloud approach, they will need to estimate the extent of services they will need to purchase up front. The Mohawk story illustrates how infrastructure can enable the strategic objectives of a firm. However, building such an infrastructure cannot come first. Firms must begin by determining a strategic vision, determining the IS architecture needed to fulfill that vision, and then making it all tangible by putting together an IS infrastructure.

This chapter examines the mechanisms by which business strategy is transformed into tangible IS architecture and infrastructure. The terms *architecture* and *infrastructure* are often used interchangeably in the context of IS. This chapter discusses how the two differ and the important role each plays in realizing a business strategy. Then this chapter examines some common architectural components for IS today.

From Vision to Implementation

As shown in Figure 6.1, architecture translates strategy into infrastructure. Building a house is similar: The owner has a vision of how the final product should look and function. The owner must decide on a strategy about where to live—in an apartment or in a house. The owner's strategy also includes deciding how to live in the house in terms of taking advantage of a beautiful view, having an open floor plan, or planning for special interests by designing such special areas as a game room, study, music room, or other amenities. The architect develops plans based on this vision. These plans, or blueprints, provide a guide—unchangeable in some areas but subject to interpretation in others—for the carpenters, plumbers, and electricians who actually construct the house. Guided by past experience and by industry standards, these builders select the materials and construction techniques best suited to the plan. The plan helps them determine where to put the plumbing and wiring, important parts of the home's infrastructure. When the process works, the completed house fulfills its owner's vision, even though he or she did not participate in the actual construction.

An IT **architecture** provides a blueprint for translating business strategy into a plan for IS. An IT **infrastructure** is everything that supports the flow and processing of information in an organization, including hardware, software, data, and network components. It consists of components, chosen and assembled in a manner that best suits the

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

126 Architecture and Infrastructure

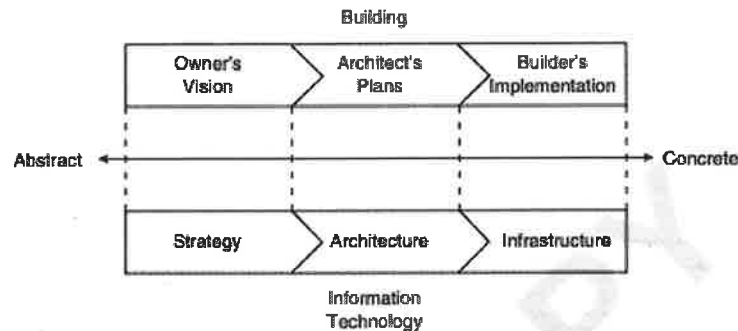


FIGURE 4.1 From the abstract to the concrete—building versus IT.

plan and therefore best enables the overarching business strategy.³ Infrastructure in an organization is similar to the beams, plumbing, and wiring in a house; it's the actual hardware, software, network, and data used to create the information system.

The Manager's Role

Even though he or she is not drawing up plans or pounding nails, the homeowner in this example needs to know what to reasonably expect from the architect and builders. The homeowner must know enough about architecture, specifically about styling and layout, to work effectively with the architect who draws up the plans. Similarly, the homeowner must know enough about construction details such as the benefits of various types of siding, windows, and insulation to set reasonable expectations for the builders.

Like the homeowner, managers must understand what to expect from IT architecture and infrastructure to be able to make full and realistic use of them. The manager must effectively communicate his or her business vision to IT architects and implementers and, if necessary, modify the plans if IT cannot realistically create or support those plans. Without the involvement of the manager, IT architects could inadvertently make decisions that limit the manager's business options in the future.

For example, a sales manager for a large distribution company did not want to partake in discussions about providing sales force automation systems for his group. He felt that a standard package offered by a well-known vendor would work fine. After all, it worked for many other companies, he rationalized, so it would be fine for his company. No architecture was designed, and no long-range thought was given to how the application might support or inhibit the sales group. After implementation, it became clear that the application had limitations and did not support the type of sales process in use at this company. He approached the IT department for help, and in the discussions that ensued, he learned that earlier infrastructure decisions now made it prohibitively expensive to implement the capability he wanted. Involvement with earlier decisions and the ability to convey his vision of what the sales group wanted to do might have resulted in an IT infrastructure that provided a platform for the changes the manager now wanted to make. Instead, the infrastructure lacked an architecture that met the business objectives of the sales and marketing departments.

The Leap from Strategy to Architecture to Infrastructure

The huge number of IT choices available coupled with the incredible speed of technology advances makes the manager's task of designing an IT infrastructure seem nearly impossible. However, in this chapter, the task is broken down into two major steps: first, translating strategy into architecture and second, translating architecture into

³ Gordon Hay and Rick Muñoz, "Establishing an IT Architecture Strategy," *Information Systems Management* 14, no. 3 (Summer 1997), 67–69.

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

infrastructure. This chapter describes a simple framework to help managers sort through IT issues. This framework stresses the need to consider business strategy when defining an organization's IT building blocks. Although this framework may not cover every possible architectural issue, it does highlight major issues associated with effectively defining IT architecture and infrastructure.

From Strategy to Architecture

The manager must start out with a strategy and then use the strategy to develop more specific goals as shown in Figure 6.2. Then detailed business requirements are derived from each goal. In the Mohawk case, the business strategy was to integrate its own product offerings with those from partners and to present the larger product line directly to a large number of customers as well as an expanded list of wholesalers. The business requirements were to integrate the disparate functionality into a modular, flexible system. By outlining the overarching business strategy and then fleshing out the business requirements associated with each goal, the manager can provide the architect with a clear picture of what IS must accomplish and the governance arrangements needed to ensure their smooth development, implementation, and use. The governance arrangements specify who in the company retains control of and responsibility for the IS. Preferably this is somebody in upper management.

Of course, the manager's job is not finished here. Continuing with Figure 6.2, the manager must work with the IT architect to translate these business requirements into a more detailed view of the systems requirements, standards, and processes that shape an IT architecture. This more detailed view, the architectural requirements, includes consideration of such things as data and process demands as well as security objectives. These are the architectural requirements. The IT architect takes the architectural requirements and designs the IT architecture.

From Architecture to Infrastructure

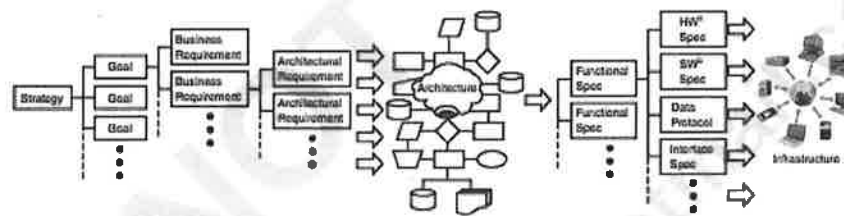
Mohawk's decision to use a service-oriented architecture led to the design of a number of services and composite applications. This illustrates the next step, translating the architecture into infrastructure. This task entails adding yet more detail to the architectural plan that emerged in the previous phase. Now the detail comprises actual hardware, data, networking, and software. Details extend to location of data and access procedures, location of firewalls, link specifications, interconnection design, and so on. This phase is also illustrated in Figure 6.2 where the architecture is translated into functional specifications. The functional specifications can be broken down into hardware specifications, software specifications, storage specifications, interface specifications, network specifications, and so on. Then decisions are made about how to implement these specifications: what hardware, software, storage, interface, network, and so forth to use in the infrastructure.

When we speak about infrastructure, we are referring to more than the components. Plumbing, electrical wiring, walls, and a roof do not make a house. Rather, these components must be assembled according to the blueprint to create a structure in which people can live. Similarly, hardware, software, data, and networks must be combined in a coherent pattern to have a viable infrastructure. This infrastructure can be considered at several levels. At the most global level, the term may be focused on the enterprise and refer to the infrastructure for the entire organization. The term may also focus on the interorganizational level by laying the foundation for communicating with customers, suppliers, or other stakeholders across organizational boundaries. Sometimes *infrastructure* refers to those components needed for an individual application. When considering the structure of a particular application, it is important to consider databases and program components, as well as the devices and operating environments on which they run.

Often when referring to an infrastructure, the underlying computer system is called the platform. The term has been used in a variety of ways: to identify the hardware and operating system of a computer, such as Microsoft Windows, Apple OSX, or Linux, or smartphone and tablet operating systems, such as Android and iOS. Vendors need to provide an entirely separate version of their software on each chosen platform, and they often have tools that allow their programs to produce, nearly automatically, versions that run on multiple platforms.

A platform can also refer to a firm's collection of cloud-based, modular tools as the example from Mohawk illustrated. Such platforms use open standards for easy "plugging-in" of components, enabling "mashing-up" of a

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



* Hardware
* Software

FIGURE 6.2 From strategy to architecture to infrastructure.

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

variety of resources at once. Google Maps is an excellent example of a standardized resource that can be accessed by any platform that provides the proper requests.

Framework for the Infrastructure and Architecture Analysis

When developing a framework for transforming business strategy into architecture and then into infrastructure, these basic components should be considered:

- **Hardware:** The physical components that handle computation, storage, or transmission of data (e.g., personal computers, servers, mainframes, hard drives, RAM, fiber-optic cabling, modems, and telephone lines).
- **Software:** The programs that run on the hardware to enable work to be performed (e.g., operating systems, databases, accounting packages, word processors, sales force automation, and enterprise resource planning systems). Software is usually divided into two groups: **system software**, such as Microsoft Windows, Apple OSX, and Linux, and **applications**, such as word processors, spreadsheets, and digital photo editors. System software is often referred to as a **platform** because application software runs upon it, sometimes only on a particular version.
- **Network:** Software and hardware components for local or long-distance networking. Local networking components include switches, hubs, and routers; long-distance networking components include cable, fiber, and microwave paths for communication and data sharing. All work according to a common protocol, most often Internet protocol (IP). Some networks are private, requiring credentials to connect. Others, like the Internet, are public.
- **Data:** The electronic representation of the numbers and text. Here, the main concern is the quantity and format of data and how often it must be transferred from one piece of hardware to another or translated from one format to another.

The framework that guides the analysis of these components was introduced in the first chapter in Figure 1.6. This framework is simplified to make the point that initially understanding an organization's infrastructure is not difficult. Understanding the technology behind each component of the infrastructure and the technical requirements of the architecture is a much more complex task. The main point is that the general manager must begin with an overview that is complete and that delivers a big picture.

This framework asks three types of questions that must be answered for each infrastructure component: what, who, and where. The "what" questions are those most commonly asked and that identify the specific type of technology. The "who" questions seek to understand what individuals, groups, and departments are involved. In most cases, the individual user is not the owner of the system or even the person who maintains it. In many cases, the systems are leased, not owned, by the company, making the owner a party completely outside the organization. In understanding the infrastructure, it is important to get a picture of the people involved. The third set of questions addresses "where" issues. With the proliferation of networks, many IS are designed and built with components in multiple locations, often even crossing oceans. Learning about infrastructure means understanding where everything is located.

We can expand the use of this framework to also understand architecture. To illustrate the connections between strategy and systems, the table in Figure 6.3 has been populated with questions that typify those asked in addressing architecture and infrastructure issues associated with each component.

The questions shown in Figure 6.3 are only representative of many that would need to be addressed; the specific questions depend on the business strategy the organizations are following. However, this framework can help IT staff ask managers to provide further information as they seek to translate business strategy into architecture and ultimately into infrastructure in their organizations. The answers derived with IT architects and implementers should provide a robust picture of the IT environment. That means that the IT architecture includes plans for the data and information, the technology (the standards to be followed and the infrastructure that provides the foundation), and the applications to be accessed via the company's IT system.

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

130 Architecture and Infrastructure

Component	What		Who		Where	
	Architecture	Infrastructure	Architecture	Infrastructure	Architecture	Infrastructure
Hardware	What type of personal device will our users use?	What size hard drives do we equip our laptops with?	Who knows the most about servers in our organization?	Who will operate the server?	Does our architecture require centralized or distributed servers?	What specific computers will we put in our Tokyo data center?
Software	Does fulfillment of our strategy require ERP software?	Shall we go with SAP or Oracle applications?	Who is affected by a move to SAP?	Who will need SAP training?	Does our geographical organization require multiple database instances?	Can we use a cloud instance of Oracle for our database?
Network	How should the network be structured to fulfill our strategy?	Will a particular Cisco switch be fast enough for what we need?	Who needs a connection to the network?	Who provides our wireless network?	Will we let each user's phone be a hotspot?	Shall we lease a cable or use satellite?
Data	What data do we need for our sales management system?	What format will we store our data in?	Who needs access to sensitive data?	How will authorized users identify themselves?	Will backups be stored on-site or off-site?	Will data be in the cloud or in our data center?

FIGURE 6.3 Infrastructure and architecture analysis framework with sample questions.

Traditionally, there are three common configurations of IT architecture as shown in Figure 6.4. Enterprises sometimes like the idea of a **centralized architecture** with everything purchased, supported, and managed centrally, usually in a **data center**, to eliminate the difficulties that come with managing a distributed infrastructure. In addition, almost every sizable enterprise has a large data center with servers and/or large **mainframe** computers that support many simultaneous users. Because of that history, there are a significant number of legacy mainframe environments still in operation today. However, one large computer at the center of the IT architecture is not used as regularly today as it was in the past. Instead, many smaller computers are linked together to form a centralized IT core that operates very much like the mainframe, providing the bulk of IT services necessary for the business.

A more common configuration is a **decentralized architecture**. The hardware, software, networking, and data are arranged in a way that distributes the processing and functionality between multiple small computers, servers, and devices, and they rely heavily on a network to connect them together. Typically, a decentralized architecture uses numerous servers, often located in different physical locations, at the backbone of the infrastructure, called a **server-based architecture**.

A third increasingly common configuration is **service-oriented architecture (SOA)**, the architecture that Mohawk, in this chapter's opening case, decided to use. An example of a service is an online employment form that, when completed, generates a file with the data for use in another service. Another example is a ticket-processing service that identifies available concert seats and allocates them. These relatively small chunks of functionality are available for many applications through **reuse**. The type of software used in an SOA architecture is often referred to as **software-as-a-service**, or SaaS. Another term for these applications when delivered over the Internet is **Web services**.

A cutting-edge type of configuration is one that can allocate or remove resources by itself, referred to as a **software-defined architecture**.⁴ Two illustrations can provide an idea of this trend. The first is a true story of a

⁴ See K. Pearlson, "Software Defined Future: Instant Provisioning of IT Services," *Connect-Converge* (Fall 2014), http://connect-converge.com/issues/2014_fall/A1767E8393A03D54262BE6F0B892F986/Converge%20C2-2014-Fall.pdf (accessed August 27, 2015).

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Architecture	Description	Other Terms	When to Use?
Centralized Architecture	- A large central computer system runs all applications and stores all data. - Typically, the computer is housed in a data center and managed directly by the IT department. - Networking allows users to access remotely.	Mainframe architecture	- To make it easier to manage—all functionality is located in one place - When the business is highly centralized
Decentralized Architecture	- Computing power is spread out among a number of devices in different locations. - Servers in different locations, personal computers, laptops, smartphones, and tablets are also included. - The "client" devices can perform many of the services needed with only occasional requests to central servers for data and services.	Server-based architecture	- To modularize and address concerns about scalability - When the business is primarily decentralized
Service-Oriented Architecture (SOA)	- Software is broken down into services "orchestrated" and connected to each other. - Together those services form an application for an entire business process. - The services are often offered from multiple vendors on the Internet and are combined to form applications.	Cloud-based architecture	- To be agile—reusability and componentization can create new apps - When the business is new and rapid app design is important
Software-Defined Architecture	- Infrastructure reconfigures based on load or time of day. - Infrastructure can be reconfigured autonomously based on rules.	Software-defined network, network virtualization	- When resources need to be flexible and reconfigured often - When usage varies dramatically depending on time of day

FIGURE 4A Common architectures.

company selling 10 bird baths per month. It had a Web site for its small family business. For a while, the site was adequate for its needs. However, when Oprah Winfrey featured the company's high-quality designs on her show, the number of monthly orders jumped to 80,000. Fortunately, the firm's IT consultants were able to create a software-defined network that adapted to the increase in orders. It was able to sense a change in the volume of orders and allocate additional resources such as storage and processing power to keep the Web site working. A typical hosting provider would have treated a monthly 8,000-fold volume increase as an attack and would shut down the site to protect it. Also, a typical provider would not have enough storage allocated for the orders. The software-defined network saved thousands of sales (and hundreds of thousands of dollars) from being lost.

Sometimes software-defined networks can even change the architecture on the fly. For example, many fast-food restaurants and coffee shops offer free WiFi to customers. This capability requires more than one connection to the Internet in very busy locations, and the shop itself needs its own secure, dedicated connection to record sales transactions and inventory updates from individual restaurant and shop operations. If that operation connection fails, a software-defined network could automatically reconfigure to switch one of the customer connections to become a substitute operations connection. Customers might find their WiFi connections to be a little slower until the situation returns to normal, but the automatic reconfiguration prevents the restaurant or shop from having to close

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

132 Architecture and Infrastructure

or revert to a very clumsy manual system. Even without a catastrophe, customer traffic on the WiFi system and the need for operations capacity can fluctuate as well. After closing, the WiFi system for customers is not needed, but during busy times, it might be saturated. When software updates are performed or large volumes of transactions are transmitted, the operations connection might be overwhelmed. Shifting resources automatically from one separate architectural component to another is a powerful way to reduce costs.

A manager must be aware of the trade-offs when considering architectural decisions. For example, decentralized architectures are more modular than centralized architectures, allowing other servers to be added with relative ease and provide increased flexibility for adding clients with specific functionality for specific users. Decentralized organizational governance, such as that associated with the networked organization structure (discussed in Chapter 3), is consistent with decentralized architectures. In contrast, a centralized architecture is easier to manage in some ways because all functionality is centralized in the main computer instead of distributed throughout all the devices and servers. A centralized architecture tends to be a better match in companies with highly centralized governance, for example, those with hierarchical organization structures. SOA is increasingly popular because the design enables large units of functionality to be built almost entirely from existing software service components. SOA is useful for building applications quickly because it offers managers a modular and componentized design and, therefore, a more easily modifiable approach to building applications. **Software-defined architectures** are even easier to manage because they self-manage many of their features. However, each self-managing feature must be imagined and defined; the systems are not autonomous beyond those features.

An example of an organization making these trade-offs is the Veterans Health Administration (VHA), a part of the Department of Veterans Affairs of the U.S. federal government.³ The organization included 14 different business units that served various administrative and organizational needs. The primary objective of the organization was to provide health care for veterans and their families. In addition, the VHA was a major contributor to medical research, allowing medical students to train at VHA hospitals. The medical centers operated independently and sometimes competed against each other. When the U.S. Congress passed an act that enabled the VHA to restructure itself from a system of hospitals to a single health care system, the IT architecture was reconfigured from a very centralized design, which enabled the Office of Data Management and Telecommunications to retain control, to a decentralized hospital-based architecture that gave local physicians and administrators the opportunity to deploy applications addressing local needs while ensuring that standards were developed across the different locations. The VA then introduced the "One-VA" architecture to unify the decentralized systems and "to provide an accessible source of consistent, reliable, accurate, useful, and secure information and knowledge to veterans and their families. . . ." Efforts were made to encrypt, secure, and account for every piece of computer hardware in the system, and a national and regional data warehouse initiative was launched to standardize business data storage and management.

Technological advances such as peer-to-peer architecture and wireless or mobile infrastructure make possible a wide variety of options. These designs can either augment a firm's existing way of operating or become its main focus. For example, a **peer-to-peer** architecture allows networked computers to share resources without needing a central server to play a dominant role. ThePirateBay.org, the Web site for sharing music, movies, games, and more, and Skype, a site for teleconferencing, texting, and telephoning, are examples of businesses that use a peer-to-peer architecture. **Wireless (mobile) infrastructures** allow communication from remote locations using a variety of wireless technologies (e.g., fixed microwave links; wireless LANs; data over cellular networks; wireless WANs; satellite links; digital dispatch networks; one-way and two-way paging networks; diffuse infrared, laser-based communications; keyless car entry; and global positioning systems).

Web-based and cloud architectures locate significant hardware, software, and possibly even data elements on the Internet. Web-based architectures offers greater flexibility when used as a source for **capacity-on-demand**, or the availability of additional processing capability for a fee. IT managers like the concept of capacity on demand to help manage peak processing periods when additional capacity is needed. It allows them to use the Web-available capacity as needed, rather than purchasing additional computers to handle the larger loads.

³ Adapted from V. Venkatesh, H. Bala, S. Venkatraman, and J. Bates, "Enterprise Architecture Maturity: The Story of the Veterans Health Administration," *MIS Quarterly Executive* 6, no. 2 (June 2007), 79–90; and J. Walters, "IBM Transformation Series, 2009," <http://www.businessofgovernment.org/report/transforming-information-technology-department-veterans-affairs> (accessed August 27, 2015).

⁴ Venkatesh, Venkatraman, and Bates, "Enterprise Architecture Maturity," p. 86.

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

With the proliferation of smartphones and tablets, enterprises increasingly have employees who want to bring their own devices and connect to enterprise systems. Some call this **Bring Your Own Device (BYOD)**, and it raises some important managerial considerations. When employees connect their own devices to the corporate network, issues such as capacity, security, and compatibility arise. For example, many corporate applications are not designed to function on the small screen of a smartphone. Redesigning them for personal devices may require significant investment to accommodate the smartphone platform. And not all smartphone platforms are the same. Designing for an iPhone is different than for an Android phone. Even if a system were redesigned for these two platforms, the resources required to maintain the system increase because each platform evolves at a different rate and the applications need to appear similar on each device. In some circles, the drive to port applications to personal devices and the ensuing issues to make them work is referred to as the **consumerization of IT**.

Consumerization of IT is a growing phenomenon. Not only do employees want to use their own devices to access corporate systems but also customers increasingly expect to access company systems from their mobile devices. Making applications robust yet simple enough for customers to use from virtually any mobile device over the Web is a challenge for many information systems departments. Companies such as Good Technology have been created to provide services that allow enterprise employees to connect, communicate, and collaborate using their own devices, supplementing the IT organization's ability to meet this new demand. Websites are designed with the philosophy of "responsive design," permitting them to adapt to screens of any size.

From Strategy to Architecture to Infrastructure: An Example

This section⁷ considers a simple example to illustrate the process of converting strategy to architecture to infrastructure: We introduce GiantCo.com, a fictitious competitor of Amazon and Wal-Mart, which sells a wide variety of products online.

Define the Strategic Goals

The managers at GiantCo.com recognize that they have a large amount of competition, so they have decided to try to provide outstanding customer service. In fact, their strategy is to become highly customer focused. Among their immediate strategic goals are the following:

- To increase the period of a money-back guarantee from one week to a month
- To provide cross-selling opportunities by temporarily discounting accessories or items that complement those purchased within the previous year
- To provide a return shipping label with every purchase
- To decrease out-of-stock occurrences by 20%
- To answer emails within 24 hours

Translate Strategic Goals to Business Requirements

To keep things simple, consider more closely only the first two of GiantCo.com's strategic goals: to increase the period of a money-back guarantee from one week to a month and to suggest goods that complement all those sold to a customer in the past year. How can GiantCo.com's architecture enable this goal? Its goal must be translated into business requirements. A few of the business requirements that address these two goals are to track

- At least a year's worth of sales for all customers
- All refunds provided to customers

⁷ Only a few questions raised from the framework are provided; a comprehensive, detailed treatment of this situation would require more information than provided in this simple example.

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

134 Architecture and Infrastructure

- Return patterns by customer to detect excesses
- Sales of complementary goods to provide advice for future potential purchasers

Translate Business Requirements into Architecture

To support the business requirements, architectural requirements are specified that dictate the architecture to be established. One major component of the architecture deals with how to obtain, store, and use data to support the business requirements.

The database needs to store the sales data for all customers for more than an entire year. The data can be used for many purposes, including summarizing for an annual report and identifying whether customers who wish to return goods are within the 30-day period. It also provides the foundation for suggesting complementary goods when coupled with data pinpointing goods that are related. As customers use the Web site, the sales data can be very useful for their own decision making.

Translate Architecture to Infrastructure

With the architecture goals in hand, the framework presented in Figure 6.2 outlines how to build the infrastructure. The architecture outlines the functions needed by the infrastructure, enabling a functional specification to be created. Those specs are then translated into hardware, software, data protocols, interface designs, and other components that will make up the infrastructure. For GiantCo.com's database, the functional specification would include details such as how big it should be, how fast data access should be, what the format of the data will be, and more. These functional specifications then help narrow the technical specifications, which answer these questions. For example, after considering the current customer base and forecasts for growth, GiantCo.com's database might need the following:

- Sample functional specifications for a year's worth of activity
 - Space to fit transaction data for 22,500 customers who purchase 25 items a year on average with 30 facts (date, price, quantity, item number, customer number, address shipped, credit card billed, and so on) recorded for each. On average, each fact occupies 10 characters of storage.
 - Ability to insert 1,070 records per minute. One server can handle one update per second, or 60 per minute, suggesting the need for 18 servers to handle online sales. Accounting information will be placed on its own server. That totals 168,750,000,000 characters of storage for the year, indicating that 200 gigabytes will be needed for this information alone. An analysis of vendors' products and pricing indicates that one terabyte is considered more than adequate for each server given that 18 will be purchased.
 - Software to do the required tracking for suggesting complementary goods because the current system does not have that functionality.
- Hardware specifications
 - One terabyte RAID (redundant array) level 3 hard drive space.
 - Nineteen 3-gigahertz Core 2 duo servers.
- Software specifications
 - Apache operating system.
 - My SQL database.

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Hardware	Software	Network	Data
19 servers: • 18 for sales • 1 for accounting LaCie 10-GB Thunderbolt RAID hard drive storage system	ERP system with modules for • Sales • Accounting • Inventory Enterprise application integration (EAI) software Apache operating system MySQL database software	• Cable modem to ISP • Dial-up lines for backup • Cisco routers, hubs, and switches • Firewalls from CheckPoint	Database • Sales • Inventory • Accounting • Complementary items

FIGURE 6.5 GiantCo.com's Infrastructure components.

Additional technical specifications would be created until the entire infrastructure is designed. Then GiantCo.com's IT department is ready to pick specific hardware, software, network, data, etc., to put into its infrastructure.

Figure 6.5 lists possible infrastructure components needed by GiantCo.com.

Architectural Principles

Any good architecture is based on a set of principles, or fundamental beliefs about how the architecture should function. Architectural principles must be consistent with both the values of the enterprise as well as with the technology used in the infrastructure. The principles are designed by considering the key objectives of the organization and then translated into principles to apply to the design of the IT architecture. The number of principles vary widely, and there is no set list of what must be included in a set of architectural principles. However, a guideline for developing architectural principles is to make sure they are directly related to the operating model of the enterprise and IS organization. Principles should define the desirable behaviors of the IT systems and the role of the organization(s) that support it. A sample of architectural principles is shown in Figure 6.6.

Principle	Description of What the Architecture Should Promote
Ease of use	Ease of use in building and supporting the architecture and solutions based on the architecture
Single point of view	A consistent, integrated view of the business regardless of how it is accessed
Buy rather than build	Purchase of applications, components, and enabling frameworks unless there is a competitive reason to develop them internally
Speed and quality	Acceleration of time to market for solutions while still maintaining required quality levels
Flexibility and agility	Flexibility to support changing business needs while enabling evolution of the architecture and the solutions built on it
Innovation	Incorporation of new technologies, facilitating innovation
Data security	Data protection from unauthorized use and disclosure
Common data vocabulary	Consistent definitions of data throughout the enterprise, which are understandable and available to all users
Data quality	Accountability of each data element through a trustee responsible for data quality
Data asset	Management of data like other valuable assets

FIGURE 6.6 Sample architectural principles.

Source: Adapted from examples of IT architecture from IBM, The Open Group Architecture Framework, the U.S. Government, and the State of Wisconsin.

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

136 Architecture and Infrastructure

Enterprise Architecture

Many companies apply even more complex and comprehensive frameworks than those described earlier for developing an IT architecture and infrastructure than those described earlier. They employ an **enterprise architecture (EA)**, or the "blueprint" for all IS and their interrelationships in the firm. EA is the term used for the organizing logic for the entire organization. It often specifies how information technologies support business processes. EA differs from an IT architecture in its level of analysis, although it shares some design principles of the lower-level architectures. It identifies the core processes of the company and how they will work together, how the IT systems will support the processes, the standard technical capabilities and activities for all parts of the enterprise, and guidelines for making choices. As experts Jeanne Ross, Peter Weill, and David Robertson describe in their book, *Enterprise Architecture as Strategy*,

Top-performing companies define how they will do business (an operating model) and design the processes and infrastructure critical to their current and future operations (enterprise architecture). . . . Then these smart companies exploit their foundation, embedding new initiatives and using it as a competitive weapon to seize new business opportunities.⁸

The components of an enterprise architecture typically include four key elements:

- **Core business processes:** The key enterprise processes that create the capabilities the company uses to execute its operating model and create market opportunities
- **Shared data:** The data that drive the core processes
- **Linking and automation technologies:** The software, hardware, and networking technologies that provide the links between applications (applications themselves are part of the IT architecture, but the way applications link together is part of the bigger picture of the enterprise architecture)
- **Customer groups:** Key customers to be served by the architecture⁹

One example of an enterprise architecture framework is the **TOGAF** (The Open Group Architecture Framework).¹⁰ TOGAF includes a methodology and set of resources for developing an enterprise architecture. It is based on the idea of an open architecture, one whose specifications are public (as compared to a proprietary architecture whose specifications are not made public). It is based on the U.S. Department of Defense frameworks and has been developing and continuously evolving since the mid-1990s. It provides a practical, standardized methodology (called *Architecture Development Methodology*) to successfully implement an enterprise architecture for an organization. Although there is no well-accepted standard for enterprise architecture, architects who understand and use TOGAF speak a common language and use the same basic framework and processes to build their company's IS architecture. TOGAF is designed to translate strategy into architecture and then into a detailed infrastructure; however, it supports a much higher level of architecture that includes more components of the enterprise.¹¹

Another example of enterprise architecture frameworks is the **Zachman framework**, which determines architectural requirements by providing a broad view that helps guide the analysis of the detailed view. This framework's perspectives range from the company's scope, to its critical models and, finally, to very detailed representations of the data, programs, networks, security, and so on. The models it uses are the conceptual business model, the logical system model, and the physical technical model.¹²

Enterprise architectures mature as firms invest resources in technologies that support their strategy. Jeanne Ross¹³ theorized that enterprise architecture moves from compartmentalized "silos" to standardized technologies to enterprisewide software to business modularity. A recent study¹⁴ shows a dramatic increase in perceived IT effectiveness as the architecture matures through those four stages.

⁸ Jeanne W. Ross, Peter Weill, and David C. Robertson, *Enterprise Architecture as Strategy* (Boston, MA: Harvard Business School Press, 2006), viii–ix.

⁹ Ibid., 50–52.

¹⁰ The Open Group, <http://www.opengroup.org>.

¹¹ For more information on the TOGAF framework, visit the Open Group's Web site at www.opengroup.org/togaf.

¹² For more information on the Zachman framework, visit Zachman International's Web site at www.zachman.com.

¹³ J. W. Ross, "Creating a Strategic IT Architecture Competency: Learning in Stages," *MIS Quarterly Executive* 2, no. 1 (2003), 31–43.

¹⁴ Randy V. Bradley, Renée M. E. Pratt, Terry Anthony Byrd, and Lakisha L. Simmons, "The Role of Enterprise Architecture in the Quest for IT Value," *MIS Quarterly Executive* 10, no. 2 (2011), 19–27.

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Because enterprise architecture is more about how the company operates than how the technology is designed, building an EA is a joint exercise to be done with business leaders and IT leaders. IT leaders cannot and should not do this alone. Because virtually all business processes today involve some component of IT, the idea of trying to align IT with business processes would merely automate or update processes already in place. Instead, business processes are designed concurrently with IT systems. The Mohawk case at the beginning of this chapter illustrates this very well; if Mohawk had simply continued its existing business processes or had made them faster with newer technology, its profitability would have merely continued to decline. The company was able to reverse this trend only by redesigning or redirecting its business processes, an effort that was enabled by IT.

As Mohawk found, building an enterprise architecture is more than just linking the business processes to IT. It starts with organizational clarity of vision and strategy and places a high value on consistency in approach as a means of optimal effectiveness. The consistency manifests itself as some level of standardization—standardization of processes, deliverables, roles, and/or data. Every EA has elements of all these types of standardization; however, the degree and proportion of each vary with organizational needs, making it dynamic. A good enterprise architect understands this and looks for the right blend for each activity the business undertakes. That means that because organizational groups and individuals are resources for business processes, the organizational design decisions should be part of the enterprise architecture. However, this is a sophisticated approach, and new enterprise architects often seek to put more rigid standards in place and do not attempt to tackle the more complex organizational design issues.

Barclay's Bank,¹⁵ which services more than 48 million customers worldwide, had an IT architecture that included more than 2,000 applications and spent in excess of £1 billion annually on IT. The resulting complexity was managed with an EA that specified frameworks, tools, and processes that created a common language and format. The EA governance model dictated that both business and technology executives sign off on projects to ensure accountability and ownership. Roadmaps helped clarify the enterprise architecture design and direction, which informed planning and portfolio management and created a common vision and a repeatable mechanism for future investments. The EA ensured appropriate linkages between IT investment and business needs.

Virtualization and Cloud Computing

Physical corporate data centers are rapidly being replaced by virtual infrastructure called **virtualization**. *Virtual infrastructure* originally meant one in which software replaced hardware in a way that a "virtual machine" or a "virtual desktop system" was accessible to provide computing power. Typically, computing capabilities, storage, and networking are provided by a third party or group of vendors, usually over the Internet or through a private network. In most virtual architectures, the five core components available virtually are servers, storage, backup, network, and disaster recovery. Virtualizing the desktop is a common virtualization application. In a virtualized desktop, the user's device locally accesses desktop software on a remote server, essentially separating the operating system from the applications. Virtualization is a useful way to design architecture because it enables resources to be shared and allocated as needed by the user and makes maintenance easier because resources are centralized.

Cloud computing is another term used to describe an architecture based on services provided over the Internet. It is based on the concept of a virtual infrastructure. Entire computing infrastructures are available "in the cloud." Using the cloud to provide infrastructure means that the cloud is essentially a large cluster of virtual servers or storage devices. This is called *infrastructure as a service (IaaS)*.

In addition to IaaS, software as a service (SaaS) and platform as a service (PaaS) are typical services found in cloud computing. These are described more fully in Chapter 10. Using the cloud for a platform means that the manager will use an environment with the basic software available, such as Web software, applications, database, and collaboration tools. Using the cloud for an entire application generally means that the software is custom designed or custom configured for the business but resides in the cloud.

¹⁵ Adapted from Phil LeClare and Eric Knorr, "The 2010 Enterprise Architecture Awards" (September 10, 2010), <http://www.infoworld.com/d/architecture/the-2010-enterprise-architecture-awards-823> (accessed August 27, 2015).

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

138 Architecture and Infrastructure

Consumers of cloud computing purchase capacity on demand and are not generally concerned with the underlying technologies. It's the next step in **utility computing**, or purchasing any part of the consumers' storage or processing infrastructure they need when they need it. Much like the distribution of electricity, the vision of utility computing is that computing infrastructure would be available when needed in as much quantity as needed. When the lights and appliances are turned off in a home, the electricity is not consumed. Ultimately, the customer is billed only for what is used. In utility computing, a company uses a third-party infrastructure to do their processing or transactions and pay only for what they use. And as in the case of the electrical utility, the economies of scale enjoyed by the computing utility enable very attractive financial models for their customers. As the cost of connectivity falls, models of cloud computing emerge.

Salesforce.com, Facebook, Gmail, Windows Azure, Apple iTunes, and LinkedIn are examples of applications in the cloud. Users access LinkedIn through the Web and build networks of business professionals on the site. But LinkedIn provides additional services, such as linking a user's blog to her or his profile, sharing and storing documents among group's members, and accessing applications such as GoodReads to see what network peers are reading and TripIt to learn about their travel plans.

Benefits of virtualization and cloud computing are many. Businesses that embrace a virtual infrastructure can consolidate physical servers and possibly eliminate many of them, greatly reducing the physical costs of the data center. Fees can be based on transaction volumes rather than large up-front investments. There is no separate cost for upgrade, maintenance, and electricity. Nor is there a need to devote physical space or to guess how many storage servers are required. Typically, the network is much simpler, too, because the virtual infrastructure mainly requires Internet connections for all applications and devices.

But the biggest benefit of virtualization and cloud computing is the speed at which additional capacity, or provisioning, can be done. In a traditional data center, additional capacity is often a matter of purchasing additional hardware, waiting for its delivery, physically installing it, and ensuring its compatibility with the existing systems. It can take weeks. In a virtual infrastructure, the nature of the architecture is dynamic by design, making adding capacity relatively easy and quick.

For example, *The New York Times* decided to make all public domain articles from 1851 to 1922 available on the Internet. To do that, the company decided to create PDF files of all the articles from the original papers in its archives. This required scanning each column of the story, creating a series of graphic pictures of the scanned image, and then cobbling them together to create the single PDF for each story. This was a lot of work and required significant computing power. Once this batch of articles was converted and added to the company's existing library, the 11 million *New York Times* stories from 1851 to 1989 were accessible on the Internet.

The manager of this project had an idea to use the cloud. He selected a service offered by Amazon.com, Amazon EC2, wrote some code to do the project he envisioned, and tested it on the Amazon servers. He used his credit card to charge the \$240 it cost him to do this conversion. He calculated it would have taken him at least a month to do the conversion if he used only the few servers available to him in *The New York Times* network. However, using the Amazon cloud services, he was able to use a virtual server cluster of 100 servers, and it took just under 24 hours to process the entire 11 million articles.¹⁶

But managers considering virtualization and cloud computing must also understand the risks. First is the dependence on the third-party supplier. Building applications that work in the cloud may mean retooling existing applications for the cloud's infrastructure. The dominant vendor, as of the writing of this text, is VMware, a company that offers software for workstations, virtual desktop infrastructures, and servers. However, because there are no standards for virtual infrastructure, applications running on one vendor's infrastructure may not port easily to another vendor's environment.

Architectures are increasingly providing cloud computing and virtualization as alternatives to in-house infrastructures. As coordination costs drop and new platforms in the cloud are introduced, cloud computing utilization will increase.

¹⁶ Galen Gruman, "Early Experiments in Cloud Computing," *InfoWorld* (April 7, 2008), <http://www.infoworld.com/article/2649759/operating-systems/early-experiments-in-cloud-computing.html> (accessed July 28, 2015); Derek Gottfrid, "Self-Service, Prorated Supercomputing Fun!" (November 1, 2007), <http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/> (accessed July 28, 2015).

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Other Managerial Considerations

The infrastructure and architecture framework shown in Figure 6.3 guides the manager toward the design and implementation of an appropriate infrastructure. Defining an IT architecture that fulfills an organization's needs today is relatively simple; the problem is that by the time it is installed, those needs can change. The primary reason to base an architecture on an organization's strategic goals is to allow for inevitable future changes—changes in the business environment, organization, IT requirements, and technology itself. Considering future impacts should include analyzing the existing architecture, the strategic time frame, technological advances, and financial constraints.

Understanding Existing Architecture

At the beginning of any project, the first step is to assess the current situation. Understanding existing IT architecture allows the manager to evaluate the IT requirements of an evolving business strategy against current IT capacity. The architecture, rather than the infrastructure, is the basis for this evaluation because the specific technologies used to build the infrastructure are chosen based on the overall plan, or architecture. As previously discussed, these architectural plans support the business strategy. Assuming that some overlap is found, the manager can then evaluate the associated infrastructure and the degree to which it can be utilized going forward.

Relevant questions for managers to ask include the following:

- What IT architecture is already in place?
- Is the company developing the IT architecture from scratch?
- Is the company replacing an existing architecture?
- Does the company need to work within the confines of an existing architecture?
- Is the company expanding an existing architecture?

Starting from scratch allows the most flexibility in determining how architecture can enable a new business strategy, and a clean architectural slate generally translates into a clean infrastructure slate. However, planning effectively even when starting from scratch can be a challenge. For example, in a resource-starved start-up environment, it is far too easy to let effective IT planning fall by the wayside. Sometimes the problem is less a shortcoming in IT management and more one of poorly devised business strategy. A strong business strategy is a prerequisite for IT architecture design, which is in turn a prerequisite for infrastructure design.

Of course, managers seldom enjoy the relative luxury of starting with a clean IT slate. More often, they must deal in some way with an existing architecture, infrastructure, and legacy systems already in place. In this case, they encounter both opportunity—to leverage the existing architecture and infrastructure and their attendant human resource experience pool—and the challenge of overcoming or working within the old system's shortcomings. By implementing the following steps, managers can derive the most value and suffer the least pain when working with legacy architectures and infrastructures.

1. *Objectively analyze the existing architecture and infrastructure:* Remember that architecture and infrastructure are separate entities; managers must assess the capability, capacity, reliability, and expandability of each.
2. *Objectively analyze the strategy served by the existing architecture:* What were the strategic goals it was designed to attain? To what extent do those goals align with current strategic goals?
3. *Objectively analyze the ability of the existing architecture and infrastructure to further the current strategic goals:* In what areas is alignment present? What parts of the existing architecture or infrastructure must be modified? Replaced?

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

140 Architecture and Infrastructure

Whether managers are facing a fresh start or an existing architecture, they must ensure that the architecture will satisfy their strategic requirements and that the associated infrastructure is modern and efficient. The following sections describe evaluation criteria including strategic time frame, technical issues (adaptability, scalability, standardization, maintainability), and financial issues.

Assessing Strategic Timeframe

Understanding the life span of an IT infrastructure and architecture is critical. How far into the future does the strategy extend? How long can the architecture and its associated infrastructure fulfill strategic goals? What issues could arise and change these assumptions?

Answers to these questions vary widely from industry to industry. Strategic time frames depend on industry-wide factors such as level of commitment to fixed resources, maturity of the industry, cyclicalities, and barriers to entry. The competitive environment has increased the pace of change to the point that requires any strategic decision be viewed as temporary.

Architectural longevity depends not only on the strategic planning horizon, but also on the nature of a manager's reliance on IT and on the specific rate of advances affecting the information technologies on which he or she depends. Today's architectures must be designed with maximum flexibility and scalability to ensure they can handle imminent business changes. Imagine the planning horizon for a dot-com company in an industry in which Internet technologies and applications are changing daily, if not more often. You might remember the importance of flexibility and agility to Mohawk's new business strategy and that the firm's IT architecture was created to support it.

Assessing Technical Issues: Adaptability

With the rapid pace of business, it is no longer possible to build a static information system to support businesses. Instead, adaptability is a core design principle of every IT architecture and one reason why cloud computing and virtualization are increasingly popular. A manager may think of technological advances as primarily affecting IT infrastructure, but the architecture must be able to support any such advance. Can the architecture adapt to emerging technologies? Can a manager delay the implementation of certain components until he or she can evaluate the potential of new technologies?

At a minimum, the architecture should be able to handle expected technological advances, such as innovations in storage capacity and computing power. An exceptional architecture also has the capacity to absorb unexpected technological leaps. Both hardware and software should be considered when promoting adaptability. For example, new Web-based applications that may benefit the corporation emerge daily. The architecture must be able to integrate these new technologies without violating the architecture principles or significantly disrupting business operations.

The following are guidelines for planning adaptable IT architecture and infrastructure. At this point, these two terms are used together because in most IT planning, they are discussed together. These guidelines are derived from work by Meta Group.¹⁷

- *Plan for applications and systems that are independent and loosely coupled rather than monolithic:* This approach allows managers to modify or replace only those applications affected by a change in the state of technology.
- *Set clear boundaries between infrastructure components:* If one component changes, others are minimally affected, or if effects are unavoidable, the impact is easily identifiable and quantifiable.
- *When designing a network architecture, provide access to all users when it makes sense to do so (i.e., when security concerns allow it):* A robust and consistent network architecture simplifies training and knowledge

¹⁷ Larry R. DeBoever and Richard D. Buchanan, "Three Architectural Sins," *CIO* (May 1, 1997), 124, 126.

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

sharing and provides some resource redundancy. An example is an architecture that allows employees to use a different server or printer if their local one goes down.

Note that requirements concerning reliability may conflict with the need for technological adaptability under certain circumstances. If the architecture requires high reliability, a manager seldom is tempted by bleeding-edge technologies. The competitive advantage offered by bleeding-edge technologies is often eroded by downtime and problems resulting from pioneering efforts with the technology.

Assessing Technical Issues: Scalability

A large number of other technical issues should also be considered when selecting an architecture or infrastructure. A frequently used criterion is scalability. To be **scalable** refers to how well an infrastructure component can adapt to increased, or in some cases decreased, demands. A scalable network system, for instance, could start with just a few nodes but could easily be expanded to include thousands of nodes. Scalability is an important technical feature because it means that an investment can be made in an infrastructure or architecture with confidence that the firm will not outgrow it.

What is the company's projected growth? What must the architecture do to support it? How will it respond if the company greatly exceeds its growth goals? What if the projected growth never materializes? These questions help define scalability needs.

Consider a case in which capacity requirements were poorly anticipated. In early 2007, an ice storm on the East Coast of the United States forced JetBlue Airlines to scramble to take care of stranded customers, grounded planes, checked luggage, and canceled flights. In the aftermath, executives told investors that the computers didn't fail. Indeed, they did not fail, but the system failed to scale as needed. The system was set up to accommodate 650 agents and was able to be increased to 950 but no more.¹⁸ It is unlikely that JetBlue or its software provider would have had to do any serious systems redesign to respond to the increase in demand; it simply needed to increase its infrastructure capacity. Ultimately, recovery from this planning failure cost JetBlue millions and even more in defending its image, which suffered severe negative word of mouth from the poor service that resulted. The company subsequently contracted with Verizon to manage its infrastructure as a way of responding to the scalability issue. JetBlue's plight underscores the importance of analyzing the impact of strategic business decisions on IT architecture and infrastructure and at least ensuring that a contingency plan exists for potential unexpected effects of a strategy change.

Assessing Technical Issues: Standardization

Another important feature deals with commonly used **standards**. Hardware and software that use a common standard as opposed to a proprietary approach are easier to plug into an existing or future infrastructure or architecture because interfaces often accompany the standard. For example, many companies use Microsoft Office software, making it an almost de facto standard. Therefore, a number of additional packages come with translators to the systems in the Office suite to make it easy to move data between systems.

Assessing Technical Issues: Maintainability

How easy is the infrastructure to maintain? Are replacement parts available? Is service available? Maintainability is a key technical consideration because the complexity of these systems increases the number of things that can go wrong, need fixing, or simply need replacing. In addition to availability of parts and service people, maintenance considerations include issues such as the length of time the system might be out of commission for maintenance,

¹⁸ Mel Duvall, "What Really Happened to JetBlue," <http://www.cioinsight.com/c/a/Past-News/What-Really-Happened-At-JetBlue> www.cioinsight.com (April 5, 2007) (accessed August 27, 2015).

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

142 Architecture and Infrastructure

how expensive and how local the parts are, and obsolescence. Should a technology become obsolete, costs for parts and expertise skyrocket. Architectures have different inherent security profiles.

Assessing Technical Issues: Security

Securing assets in a highly centralized, mainframe architecture means building protection around the centralized core. Because data and software are stored and executed on the mainframe computer, methods of protecting these assets revolve around protecting the mainframe itself. Decentralized, server-based architecture is more difficult to secure due to the dispersion of servers. Security is a matter of protecting every server instead of one centralized system. A Web-based SOA architecture that utilizes SaaS and capacity on demand raises a whole new set of security issues. The data and applications not only reside on servers in the various vendor systems around the Web, but also the linking mechanism, the network that ties the Web together, introduces another level of security concerns. Security is discussed in more detail in Chapter 7.

Assessing Financial and Managerial Issues

Like any business investment, IT infrastructure components should be evaluated based on their expected financial value. Unfortunately, payback from IT investments is often difficult to quantify; it can come in the form of increased productivity, increased interoperability with business partners, improved service for customers, or yet more abstract improvements. This suggests focusing on how IT investments enable business objectives rather than on their quantitative returns.

Still, some effort can and should be made to quantify the return on infrastructure investments. This effort can be simplified if a manager works through the following steps with the IT staff.

1. **Quantify costs:** The easy part is costing out the proposed infrastructure components and estimating the total investment necessary. Work with the IT staff to identify cost trends in the equipment the company proposes to acquire. Don't forget to include installation and training costs in the total.
2. **Determine the anticipated life cycles of system components:** Experienced IT staff or consultants can help establish life cycle trends for both a company and an industry to estimate the useful life of various systems.
3. **Quantify benefits:** The hard part is getting input from all affected user groups as well as the IT group, which presumably knows most about the equipment's capabilities. If possible, form a team with representatives from each of these groups and work together to identify all potential areas in which the new IT system may bring value.
4. **Quantify risks:** Assess any risk that might be attributable to delaying acquisition as opposed to paying more to get the latest technology now.
5. **Consider ongoing dollar costs and benefits:** Examine how the new equipment affects maintenance and upgrade costs associated with the current infrastructure.

Once this analysis is complete, the manager can calculate the company's preferred discounted cash flow (i.e., net present value or internal rate of return computation) and the payback period. Approaches to evaluating IT investments are discussed in greater detail in Chapter 8.

Applying these considerations to the fictitious GiantCo.com company, the last task is to weigh the managerial considerations against the architectural goals that were used to determine infrastructure requirements. Figure 6.7 shows how these considerations could apply to GiantCo.com's situation.

Again, note that the criteria evaluated in Figure 6.7 do not address every possible issue for GiantCo.com, but this example shows a broad sample of the issues that will arise.

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Other Managerial Considerations 143

Criteria	Architecture	Infrastructure
Strategic time frame	Indefinite: GiantCo.com's strategic goal is to be able to respond to customer needs.	NA
Technology advances	Database technology is fairly stable, but transaction capacity needs to be assessed and links with smaller suppliers and customers verified.	NA
Financial Issues		
NPV of investment	NA	GiantCo.com will analyze NPV of various hardware and software solutions and ongoing costs before investing.
Payback analysis	GiantCo.com expects the new architecture to pay for itself within three years.	Specific options will be evaluated using conservative sales growth projections to see how they match the three-year goal.
Incidental investments	The new architecture represents a moderate shift in the way GiantCo.com does business and will require some training and workforce adjustment.	Training costs for each option will be analyzed. Redeployment costs for employees displaced by any outsourcing must also be considered.
Growth requirements/scalability	Outsourcing could provide more scalability than GiantCo.com's current model, which is constrained by IT capacity. New innovations will be identified to provide scalability of volume.	The scalability required of various new hardware and software components is not significant, but options will be evaluated based on their ability to meet scalability requirements.
Standardization	NA	GiantCo.com will adopt the MySQL standard and make it a requirement of all developers for consistency.
Maintainability	The new architecture raises some maintenance issues, and new product introductions will mandate constant updates to the rules of complementary goods.	Various options will be evaluated for their maintenance and repair costs.
Staff experience	The new model will require new skills and expertise.	Current staff is not familiar with MySQL. Training and workforce adjustment will be needed. Some new staff will be hired.
Security	GiantCo.com will lock down resources for traveling personnel.	GiantCo.com will adopt a Pulse Secure VPN for securely connecting traveling personnel with network resources.

FIGURE 4.7 GiantCo.com's managerial considerations.

≡ Social Business Lens: Building Social Mobile Applications

As companies adopt social IT, they are finding that it is closely intertwined with mobile platforms. Employees want, and in some cases expect, to be able to access their social IT from their smartphones, tablets, and more. As companies look globally, in some countries the mobile screen is the only screen used.

In 2011, more than one-third of the U.S. population used the mobile Internet. In 2014, that number grew to such an extent that 52% of device owners consider smartphones and tablets the most important devices for Internet access, while only 46% consider desktops and laptops the most important devices. Tablets have surpassed all other devices in importance.

Social business requires that companies extend their architecture to include mobile functions, called *social mobile*. Social mobile functions began to take off with the widespread adoption of smartphones. The first devices combined features of a personal digital assistant with a mobile phone, giving developers the opportunity to link applications to the Web instantly. RIM's BlackBerry was one of the first to give users mobile access to communication

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

144 Architecture and Infrastructure

tools such as their e-mail. More recent devices, such as Apple's iOS, Google's Android, Microsoft's Windows Phone, Nokia's Symbian, and RIM's BlackBerry OS, use a mobile operating system.

Initial social mobile apps were social networks either ported to the mobile platform, like LinkedIn and Facebook, or designed just for the mobile platform, like Foursquare and Gowalla, social network sites linking community members who "check in" at physical locations and sometimes earn virtual rewards for doing so. Social mobile applications have extended to many other types of applications as software designers realize the large market available to them if their applications run on mobile platforms and as device users demand increasing functionality for their mobile devices.

Source: Amy Galtram, "Survey: U.S. Mobile Web Access Growing Fast" (July 8, 2010). http://articles.cnn.com/2010-07-08/tech/mobile.internet.access.pew_1_cell-phone-users-feature-phones-mobile-internet (accessed August 27, 2015); Danyl Bosonworth, "Mobile Marketing Statistics 2015," *Smart Insights* (July 22, 2015). <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/> (accessed August 27, 2015).

SUMMARY

- Strategy drives architecture, which drives infrastructure. Strategic business goals dictate IT architecture requirements. These requirements provide an extensible blueprint suggesting which infrastructure components will best facilitate the realization of the strategic goals.
- Enterprise architecture is the broad design that includes both the information systems architecture and the interrelationships in the enterprise. Often this plan specifies the logic for the entire organization. It identifies core processes, how they work together, how IT systems will support them, and the capabilities necessary to create, execute, and manage them.
- Four configurations for IT architecture are centralized, decentralized, SOA (or Web-based), and software-defined architectures. Applications are increasingly being offered as services, reducing the cost and maintenance requirements for clients. Virtualization and cloud computing provide architectures for Web-based delivery of services.
- The manager's role is to understand how to plan IT to realize business goals. With this knowledge, he or she can facilitate the process of translating business goals to IT architecture and then modify the selection of infrastructure components as necessary.
- Frameworks guide the translation from business strategy to IS design. This translation can be simplified by categorizing components into broad classes (hardware, software, network, data), which make up both IT architecture and infrastructure.
- Enterprise leaders increasingly have requests for new devices that employees want to connect to the corporate network. The consumerization of IT describes the trend to redesign corporate systems for smartphones, tablets, and other consumer-oriented devices.
- While translating strategy into architecture and then infrastructure, it is important to know the state of any existing architecture and infrastructure, to weigh current against future architectural requirements and strategic time frame, and to analyze the financial consequences of the various systems options under consideration. Systems performance should be monitored on an ongoing basis.

KEY TERMS

applications (p. 129)
architecture (p. 125)
bring-your-own-device (BYOD) (p. 133)
capacity-on-demand (p. 132)
centralized architecture (p. 130)
cloud architecture (p. 132)
cloud computing (p. 137)
consumerization of IT (p. 133)
data center (p. 130)
decentralized architecture (p. 130)
enterprise architecture (p. 136)

infrastructure (p. 125)
mainframe (p. 130)
peer-to-peer (p. 132)
platform (p. 129)
reuse (p. 130)
scalable (p. 141)
server-based architecture (p. 130)
service-oriented architecture (SOA) (p. 130)
software-as-a-service (p. 130)
software-defined architecture (p. 130)
standards (p. 141)

system software (p. 129)
TOGAF (p. 136)
utility computing (p. 138)
virtualization (p. 137)
Web-based architectures (p. 132)
Web services (p. 130)
wireless (mobile) infrastructures (p. 132)
infrastructures (p. 125)
Zachman framework (p. 136)

Printed by: shavers69@hotmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

DISCUSSION QUESTIONS

1. Think about a company you know well. What would be an example of IT architecture at that company? An example of the IT infrastructure?
2. What, in your opinion, is the difference between a decentralized architecture and a centralized architecture? What is an example of a business decision that would be affected by the choice of the architecture?
3. From your personal experience, what is an example of software as a service? Of BYOD?
4. Each of the following companies would benefit from either software-defined architecture or conventional, owned hardware and software. State which you would advise each of the following fictitious firms (plus the IRS) to adopt and explain why.
 - a. StableCo is a firm that sells industrial paper shredders. Its business has remained steady for two decades and it has a strong and diverse customer base.
 - b. DynamicCo is a fast-growing six-year old firm that has relied on three to five key wholesale customers for its entire existence. However, the list of key customers changes every year, and during two of the years, sales declined sharply.
 - c. Plastics3000 is an old, stable plastics manufacturing firm that has kept its sales steady in the face of competitors as the result of an active research and development team that uses advanced software to analyze large amounts of data to develop new compounds. Once or twice a week, office personnel complain of the network becoming very slow.
 - d. A downtown Las Vegas casino monitors each slot machine continuously for early detection of malfunctions such as winnings or losses trending beyond their threshold limits.
 - e. CallPerfect provides call center services to pharmacies. Phone calls are routed to the company after hours and messages are delivered to the pharmacy manager the next morning.
 - f. At the IRS, tax forms are available online for citizens to complete and file with the IRS electronically by April 15. A call center routes calls to agents who answer taxpayers' questions.
 - g. At LittlePeople, Inc., a day care center, parents are called using software on the administrator's computer when there is a weather emergency. The school has averaged 120 families for many years.

■ CASE STUDY 6-1 Enterprise Architecture at American Express

Enterprise architecture (EA) at American Express was the framework the organization used to align IT and the business. EA provided a common language for leaders to use to collaborate and transform the business. At American Express, enterprise architects were the change agents who streamlined processes and designed ways to more effectively do business using IT resources. In 2011, American Express was named an InfoWorld/Forrester Enterprise Architecture Award recipient for its EA practices. As American Express leaders considered new payment methods using mobile devices, the EA guided their progress.

Mobile payments were forcing the payments industry to review their practices and significantly transform the way business was done. The new business environment introduced additional complexity with the addition of new delivery channels and the need for shorter time-to-market of payment products and services. American Express's business strategy for its payments products focused on delivering a "consistent, global, integrated customer experience based on services running on a common application platform."

To achieve this goal, the EA team created reference architectures and road maps for standardized applications across the firm. This team then worked with multiple business solution delivery teams to create and manage the common application architecture and create strategies that facilitated each business's objectives. Each strategy included a road map of initiatives that included a set of actions, the metrics to evaluate the success of these actions, and the commitments IT and the businesses made to make it happen. The road map was American Express's way to standardize language, tools, life cycle management of the applications, and architecture and governance processes. The elements of the road map included technology, reference architecture, and capabilities for the business.

The next steps for American Express were to extend the road maps to cover the maturing of SOA and to develop new reference architectures and a new taxonomy to increasingly align IT with the needs of the business. As new technologies emerged and new ways of doing business over social tools created opportunities for new payment products and services, American Express expected to continually evolve its EA.

Discussion Questions

1. What are the key components of the architecture American Express has created?
2. Why was it important to standardize so much of the architecture? What are the advantages and disadvantages of a standard EA for American Express?