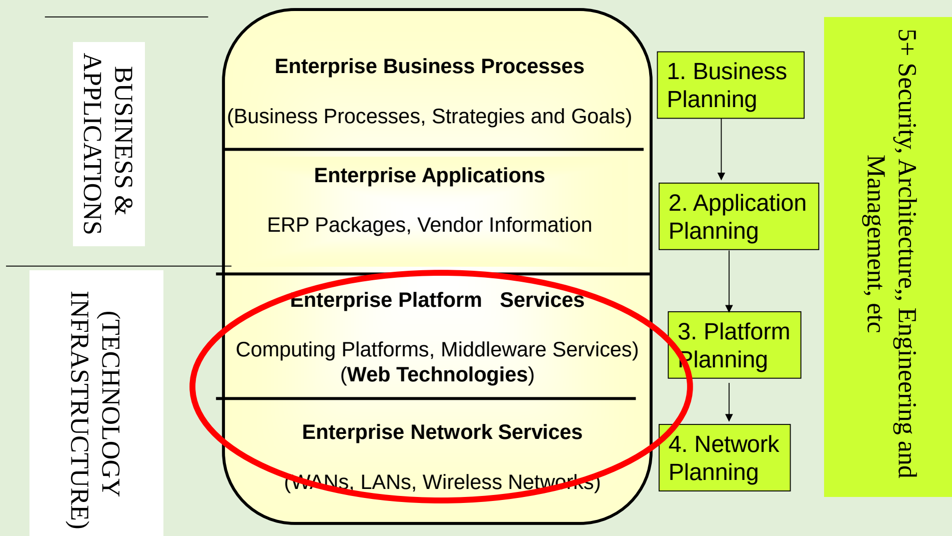
MODULE (Technology Infrastructure)

This module introduces the technical aspects of digital enterprises and presents the IT infrastructure that supports the business processes and the enterprise applications needed to support the business strategy. Specifically, this module consists of the following chapters:

Chapter 7: Overview of Enterprise Architecture and IT Infrastructure
Chapter 8: Data Bases, Big Data, and Business Intelligence
Chapter 9: Internet, Web, Mobile Communications and IoTs



7 Overview of Enterprise Architectures and IT Infrastructure

Opening Vignette: Enterprise Architecture and IT Infrastructure for a Hardware Retail Store

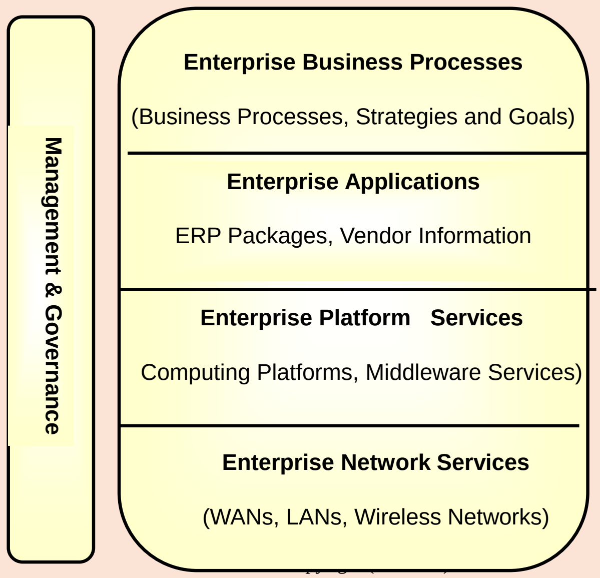
In 2010, a regional retail store in the Midwest wanted to develop a business strategy and a technical solution to double its business in three years. This raised questions such as the following:

- Where to start
- What are the business issues
- What type of business partnerships will be needed
- What are the key enabling technologies needed to support the business goals
- How do the business and technology building blocks fit together
- What methodology can be used to systematically make all decisions

After some debates, the company chose an Enterprise Architecture, shown below, that identified the key building blocks, established the interrelationships between these building blocks and served as a roadmap for this project.

The top two layers of this enterprise architecture concentrate on the business issues while the lower two layers deal with the enabling technology (IT infrastructure) considerations. IT infrastructure is the backbone of modern digital enterprises. This infrastructure consists of several building blocks that provide numerous services such as the following:

- Computing platforms that consist of operating systems and computing hardware to provide the basic scheduling and hardware services.
- Databases that house the data for day-to-day operation plus support business intelligence and management decision making.
- Networks that provide the network transport between remote parties and are responsible for routing and flow/error control support.
- Middleware that interconnects remotely located users, databases and applications – this is the *business/industry unaware* software that resides above the network but below the business applications. The Web allows access to a multitude of information sources through Web browsers.



This retail store decided to rely heavily on cloud computing instead of growing its own IT infrastructure. Cloud computing (CC) allows organizations to rent different components of IT infrastructure -- cloud computing is essentially an outgrowth of the service provider (SP) model in which all IT-related capabilities are provided "as a service" (e.g., software as a service (SaaS), platform as a service (PaaS), Infrastructure as a service (Iaas), etc.

7.1 Introduction

7.1.1 The Big Picture

Digital enterprises offer many benefits such as organizational growth and efficiency. However, they are complex and highly IT dependent. As this complexity escalates, it becomes harder and harder for IT outsiders to understand what is being achieved and how it is helping the organization advance its goals. Thus it becomes more crucial for the organizations to understand what value the increased reliance and expense on IT is providing. This is a major problem. According to META Group surveys since 2000, fewer than 30% of the IT projects meet the project goals and worse yet, senior managers from the same surveys feel that less than 20% of the IT projects "fundamentally advance the strategic goals of the enterprise".

To better understand the complex interactions between business and IT and how IT supports the business, a very simple view is presented in Figure 7-1. This view of an enterprise is presented in terms of three high level layers and their role as drivers and enablers of an enterprise:

- Business focused layer that concentrates on business strategies, services and business processes:
- Business plus technology focused layer that deals with enterprise business applications
- Technology focused layer that provides the IT (information technology) infrastructure

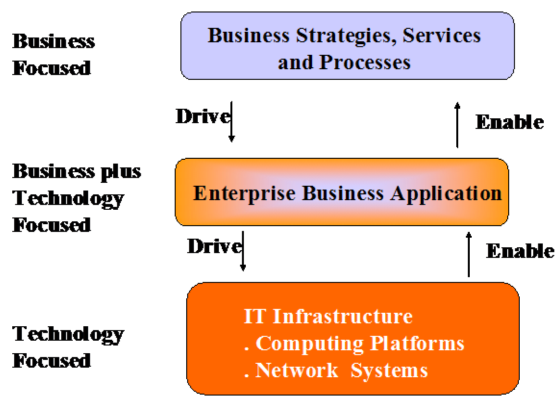


Figure 7-1: Key Components of an Enterprise

Business strategies represent the long range game plan to win in the marketplace. Business strategies align the business products/services, processes, and several other activities to survive and succeed in the marketplace**.** The term business service and business process are frequently interchanged. For the purpose of this book, we will use the following definitions;

- *Business service (BS)* is something that is delivered to the customer. For example, dry cleaning service, house renovation service and online banking represent things that are delivered to the customer, usually for a fee. A business service may be delivered to internal or external customers. For example, a payroll service is delivered to the employees of a company. Businesses usually deliver products, services or both to its customers.
- *Business process (BP)* is a collection of *activities* that are required to achieve a business goal – the goal may be a business service. At a basic level, a BP can be represented as a flowchart that specifies the orchestration of activities needed to complete the goal. For example, for a payroll *service*, several BPs have to be carried out (e.g., pay has to be computed, deductions have to be considered, overtime may need to be calculated, etc).

Thus a business service specifies what a user *receives* while a business process shows *how* the service is provided. In other words, a business service is an *external view*, while a business process defines the *internal* set of activities and their flows needed to provide a service. In many cases, BPs and BSs can be interchanged. For example, for a higher level service (e.g., customer service), many other BSs may be treated as BPs and combined with other BSs/BPs (e.g., troubleshooting, follow-up, upselling) for customer service. Unless needed, we will use BP or BS to signify the same thing. We have discussed business strategies and business processes in Chapter 4.

Enterprise business applications are the computer-based information systems that provide automated support to the business services/processes. These applications are also referred to as enterprise applications, business applications or just as applications in the literature. Whatever the name, these applications are business aware. For example, an airline reservation system contains business logic and data that is not the same as a hotel reservation system (business awareness). Business applications also provide business value to an enterprise. Obviously, an airline reservation system provides business value to the airline business. These applications use information technologies to support the enterprise and thus are enablers to the business processes. ERP (enterprise resource planning) systems, as studied previously, provide a collection of applications, integrated around a common database. We have discussed enterprise applications and ERP systems in detail in chapters 5 and 6.

The information technology (IT) infrastructure is used to build, deploy and operate the business applications. IT infrastructure, also sometimes known as computer-communication platform, consists of technologies such as computers, operating systems, networks, databases, and transaction managers. *This infrastructure enables the applications and is business unaware.* For example, the same type of networks and computers are used in airline reservation systems as well as hotel reservation systems. The best known infrastructure is the network that interconnects remote applications, databases, and users. Internet, wireless, and broadband networks are examples of vital network technologies.

An important player in modern enterprises is *middleware*, an increasingly crucial and, at the same time, bewildering component of the modern IT infrastructure. Middleware is needed to interconnect and support applications and users across a network. Middleware services typically include directories, emails, and facilities to invoke software to access and manipulate remotely located databases and applications. Middleware services are typically provided by specialized software packages (for example, a web browser is a middleware package that connects users to remotely located web resources).

As we will see, the simple but elegant view presented in Figure 7-1 will lead us to a definition of an Enterprise Architecture (i.e., Enterprise architecture (EA) = Business architecture + application architecture + technology architecture (computer platform architecture + network architecture). It will also help us to develop a basic understanding of the IT infrastructure building blocks that will be explained in the later chapters of this Module.

7.1.2 Case Study: XYZCorp Review of IT Infrastructure Architecture

XYZCorp needs to review its existing technical architecture and determine if it will enable all the services and applications to be provided by XYZCorp. The IT Planning Committee has asked you to produce a high level vision of the future infrastructure architecture. This vision will be refined later. In particular, the vision should show how the networking services, the middleware components, and other services will support the emerging enterprise applications that will also utilize mobile computing and groupware. This architecture should allow different applications and users at different sites to communicate with each other ("any data from any application anywhere in the company"). The current platforms consist of a variety of devices. The regional offices house minicomputers (mainly UNIX) which are connected to the corporate mainframe. The regional computers maintain regional inventory, customer information and prices of items sold in the region. Some regions (e.g., Atlanta and San Francisco) are very UNIX oriented. The corporate headquarters are IBM mainframe oriented and PCs are used commonly throughout the organization. The company specifically wants to know how cloud computing can be used by XYZCorp.

7.2 Enterprise Architecture and Integration Concepts

7.2.1 What is an Enterprise Architecture?

Architectures play a vital role in modern information systems because they show how the individual systems tie together to satisfy the overall requirements. For the purpose of this book, we adopt the following simple but highly operational definition of architecture:

Definition: An architecture of a system is a structure that describes three things:

- Components of the system (what are the pieces of a system?),
- Functions performed by the components (what do they do?), and
- Interfaces/interactions between the components (how do they work with each other?).

This definition is consistent with the IEEE 610.2 definition of an architecture: "The structure of the components, their properties, relationships, and the principles and guidelines governing their design and evolution over time."

An enterprise architecture (EA), based on the aforementioned definition of an architecture, shows components of an enterprise, what they do, and how they interface/interact with each other. EA is a consolidation of business and technology that can be of great value to the corporate management as well as CIOs. Figure 7-2 shows a conceptual view of an EA and its benefits in terms of four broad categories (planning, integration, security, and administration). This conceptual view is the foundation of the SPACE toolset described later. EAs, however, present many challenges to the organization. Development and maintenance of an EA can be a long and expensive undertaking. It is important to measure ROI of an EA effort. For ROI estimates, tangible measurements of the costs and benefits need to be identified. The costs of establishing an EA can be organized into distinct classes such as financial improvement, constituent services, and reduced redundancy. The benefits have been listed above. Monetizing the different types of costs and benefits facilitates the demonstration of ROI. Besides ROI, considerable effort is needed to make sure that the EA efforts are successful. For example, CIOs and CTOs can be removed from the actual development and the use of the EA document, especially in large organizations. In such cases, a group of architects, each focusing on a different dimension, generally report to a chief enterprise architect, who in turn, reports to the CIO

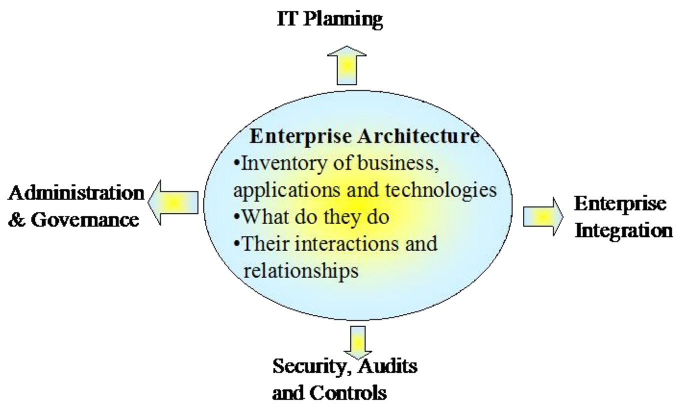


Figure 7-2: Conceptual View of an Enterprise Architecture and its Benefits

A great deal of information about enterprise architectures is available. Here are some key sources:

- The Open Group Website (http://www.opengroup.org)
- Zachman Institute for Framework Architecture (ZIFA) Website: (www.ZIFA.com) :
- EA Portal at www.enterprise-architecture.info
- Enterprise Architecture Center of Excellence (EACOE) website (http://eacoe.org)
- Gartner Group (www.gartner.org) Enterprise Architecture Practice

7.2.2 From Enterprise Architectures to Enterprise Integration

As stated previously, an enterprise architecture (EA) describes the enterprise building blocks (the business layer, the application layer, and the technology infrastructure layer), what they do and how they interface/interact with each other. How can an EA help in enterprise integration efforts? Enterprise integration means making independently designed enterprise systems work together. For enterprise integration, the goal is to provide standardized high-quality customer service across the entire firm's service channels. Multi-channel integration is critical because customers expect consistent service when they interact with a company, no matter which channel they use. Multi-channel integration is also a critical issue for any business striving to maintain its competitive advantage [Kalakota 2002a]. As we will see later, most definitions of enterprise integration touch on similar if not common concepts related to *working together*, *sharing*, *interacting*, and *collaborating*. Naturally, a well documented EA with clearly specified interfaces and interactions between various business and technical components, provides the basis for well integrated systems.

Let us use Figure 7-3 for a quick look at integrated architectures.. The objective of an *integrated* enterprise architecture is to show how *well* all the business plus technical components work together to serve the enterprise needs. Development of an integrated enterprise architecture, referred to as integrated architecture in this book for simplicity, generally starts with documenting the organization's strategy and goals.

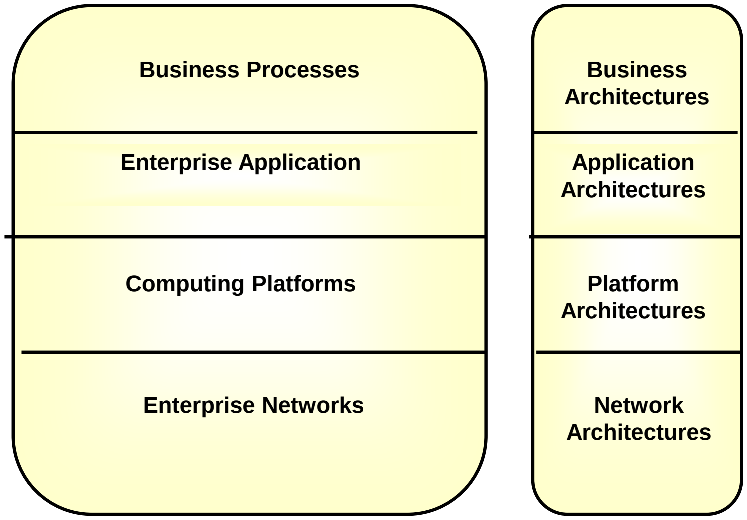


Figure 7-3: Building Blocks of Enterprise Architecture and Integration

An EA diagram such as Figure 7-3 serves as a very effective framework for enterprise wide integration -- facilitating vertical as well as horizontal integrations:

- **Vertical integrations** show a business architecture that is integrated with technology architecture (enterprise applications, platforms and networks) of a company or a division of a company. It combines business, applications and IT infrastructure components into a solution for a particular situation (typically known as a "silo"). An example of vertical integration is a supply chain management ERP system that automates all supply chain processes and operates on Linux platforms.
- **Horizontal integrations** show how processes and technologies at the same layer are integrated. For example, the integration of business processes in sales with business processes in supply chain represent a horizontal integration at business process level. Similarly, enterprise application integration represents how different ERPs within an organization seamlessly work with each other. In addition, inter-enterprise (B2B) integration architectures represent supply chain management application integration across multiple enterprises. As another example, smooth transition between wired and wireless networks (e.g., roaming support between a cellular phone, a Wi-Fi LAN and a wired corporate Intranet) represents a horizontal network integration.
- **Mixtures** represent an integrated architecture that is a combination of vertical architectures that interconnect different layers as well as the horizontal architectures at the same layers of an enterprise. In many practical cases, mergers and acquisitions lead to these integration scenarios because many organizations have vertically integrated systems but when two or more organizations merge, multiple vertical architectures need to be integrated horizontally – a real headache. This is known as the "information silo" problem.

Based on this discussion, we can introduce the following definition of *integrated* enterprise architectures:

> **Definition:** Integrated Enterprise Architecture = Horizontally Integrated Enterprise Architecture + Vertically Integrated Enterprise Architecture

7.2.3 Systems Thinking and Systems Engineering View

Simply stated, a system is a set of interrelated components which interact with one another in an organized fashion towards a common purpose. (NASA Handbook). These components may be very diverse including 5 people, processes and technologies. For example, a telemedicine system consists of human components (patients, nurses, physicians), technology components (e.g., a remote monitoring system, electronic health records (EHRs), and a computer network), and processes (e.g., policies and procedures about who can be remotely examined and who can update EHRs if needed).

Systems engineers basically build systems by selecting, acquiring, configuring, and integrating a system. The V diagram, shown in Figure 7-4, is used commonly by systems engineers to distinguish the role of component engineers who build the components from those who build “systems” by planning, architecting, integrating and managing the needed components into an operational system.

This separation of roles is becoming more important as a large number of components (e.g., mobile apps, computers, handsets, network devices, and well trained employees) are becoming available. Thus more attention is needed on systems engineering tasks than component engineering. WE will discuss this in more detail later.

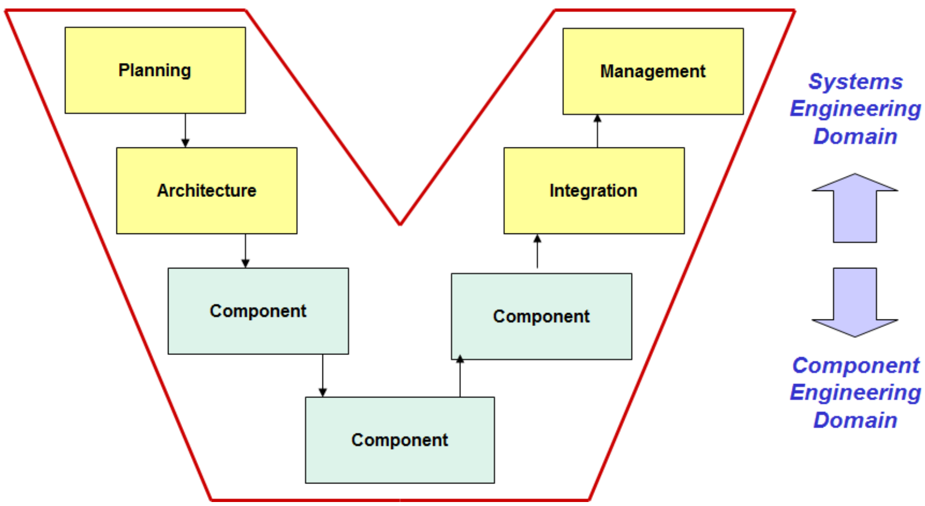


Figure 7-4: Conceptual View of Systems Engineering -- The V Diagram

Case Study: FDA Uses Enterprise Architecture to Standardize and Save

The Food and Drug Administration (FDA) needed to develop a plan for consolidation of its IT infrastructure across eight division centers and identify standard software applications to be utilized for common business needs. The FDA turned to Enterprise Architecture (EA) to serve as the methodology to achieve its desired state of efficiency and effectiveness. FDA maximized its infrastructure by replacing single-use environments with platforms shared across application system boundaries and consolidated IT operations from multiple buildings to two major locations by employing homogenous platform architecture.

EA driven IT consolidation has allowed the FDA to use fewer resources while operating more efficiently and providing better services. Key benefits of this project include: increased cost savings (over $10 million redundant IT related costs eliminated), documented and standardized business processes (over 85% of agency-wide processes are now documented), consolidation of IT infrastructure resulting in less applications (for example the number of correspondence tracking systems were reduced from 24 to 2), improved communication, and lastly improved decision making.

Source: Federal Enterprise Architecture Program Management Office, Link: http://www.whitehouse.gov/omb/egov/documents/FDA_FINAL.pdf

ITIL (Information Technology Infrastructure Library)

IT Infrastructure Library (ITIL) is a detailed framework that classifies and describes the modern IT infrastructure. ITIL is based on information gained through hands-on experience about achieving a successful governance of IT. ITIL has been developed and is maintained by the United Kingdom's Office of Government Commerce, in partnership with the IT Service Management Forum.

ITIL consists of a series of books that provide guidance on the management and services needed to support IT. ITIL books document best practices for IT Service Management and serve as the foundation for a series of certificates that are offered by the ITIL group. ITIL has been adopted by hundreds of organizations worldwide including Microsoft, IBM, Barclays Bank, HSBC, Procter & Gamble, British Airways and Hewlett Packard.

See the ITIL main website (http://www.itil-officialsite.com/home/home.asp) for additional details.

7.3 ICT Infrastructure and Technical Architecture

7.3.1 ICT infrastructure at a Glance

Technology architectures primarily consist of the IT infrastructure (i.e., middleware, networks, operating systems, and computing hardware) that enables the business and application architecture of modern enterprises. Figure 7-5 will serve as a general framework for discussion. This framework illustrates the role of the following main IT infrastructure building blocks:

- Networks that provide the network transport between remote parties and are responsible for routing and flow/error control support. The networks may be the private value added networks (VANs), Public Internet, and/or Extranets that utilize the wired or wireless transmission media.
- Computing platforms that consist of operating systems and computing hardware to provide the basic scheduling and hardware services. The computing platforms also include local system software services such as database managers, transaction managers, language translators, and utilities.
- Middleware that interconnects remotely located users, databases and applications. Middleware components are *business/industry unaware* software modules that provide a variety of services such as Web services, directory services, email, and remote data access services. The Web allows access to a multitude of resources through the Web browsers.

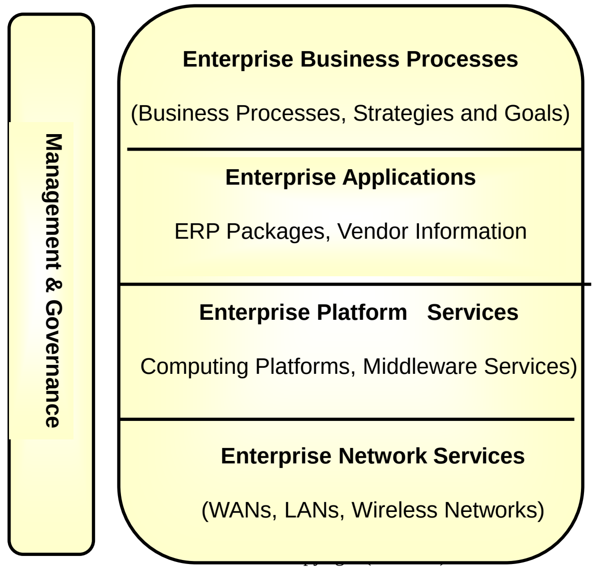


Figure 7-5: IT Infrastructure

IT planning translates the various IT infrastructure building blocks into a technical architecture that shows the main components of the IT components (e.g., computing platforms) and how they are interconnected by using wireless as well as wired network elements. Figure 7-6 shows a sample technical architecture of a small company. This view represents a typical enterprise technical architecture that shows the computers, the databases and application programs. It also shows an Intranet for internal use, an Extranet for business partners, multiple wired/wireless LANs connected to the Intranet backbone and a Public Internet connection for the customers. This technical architecture can be customized and expanded for different types of enterprises. We will introduce these building blocks briefly here and then discuss them in more detail in later chapters.

7.3.1 Computing Platforms – Operating Systems and Local Systems Software

Computing platforms, as shown in Figure 7-7, provide the basic computing hardware, operating systems, and local systems software such as database and transaction managers to support applications. We do not discuss computing hardware, but operating systems and local system software packages are reviewed briefly.

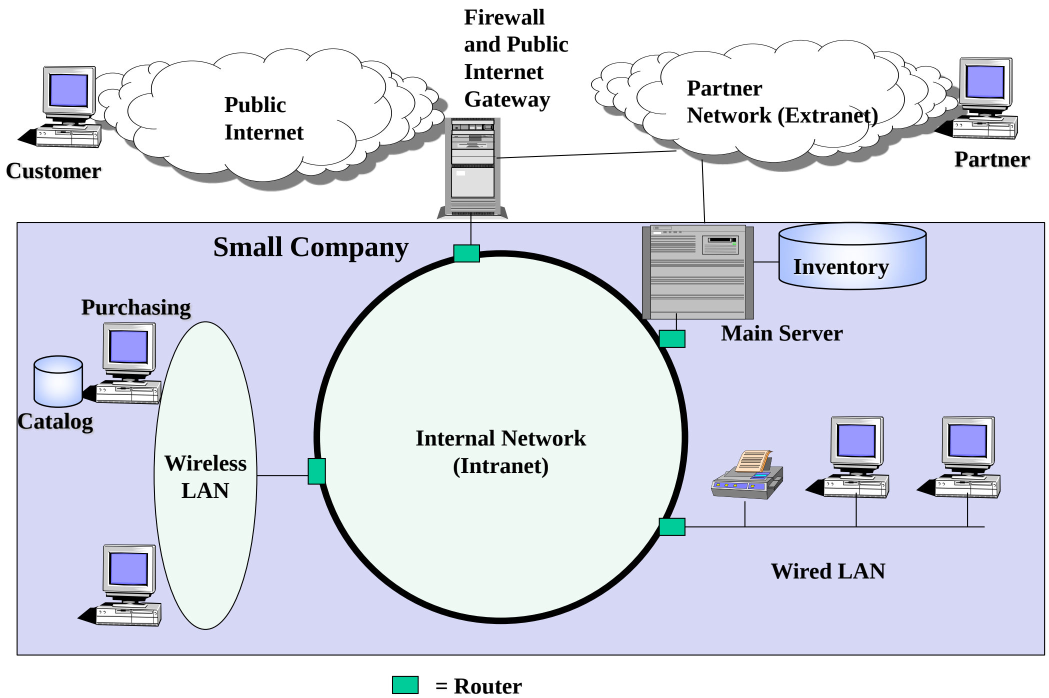


Figure 7-6: *Example of a Technical Architecture for a Small Company*

7.3.1.1 Computing Hardware Platforms

Computing hardware platforms include client machines (desktop PCs, laptops, and mobile computing devices such as iPhones, Androids, and notebooks) and server machines.

Client machines are typically used for one user. These machines are becoming more powerful in terms of more processing speeds and disk space with smaller footprint.

Servers are more powerful than the client machines and are designed for several users (in the range of 100-500 users). Most modern servers use Blades – ultrathin computers consisting of a circuit board with processors, memory, and network connections that are stored in racks. Blade servers take up less space than traditional box-based servers. Secondary storage may be provided by a hard drive in each blade server or by external mass-storage drives.

Mainframes are large scale machine that are designed for thousands of users simultaneously. IBM has dominated this market since the 1970s. Mainframes are in fact giant servers for massive enterprise networks and corporate Web sites. A single IBM mainframe can run up to 20,000 instances of Linux or Windows server software and can replace thousands of smaller blade servers. At present, small to medium businesses can be easily supported by one server connected to several client machines. However, for large corporations, mainframes are used commonly for corporate business processing.

7.3.1.2 Operating Systems

Simply stated, an operating system is the computer system's chief manager. As shown in Figure 7-7, the operating system surrounds the computing hardware and all other software (local system software such as database managers, middleware, application software, and the user interfaces) rely on the operating system. It decides which hardware resources will be used, which programs will be run, and the order in which activities will take place. Technically, an operating system is a program, or a collection of programs, which allocates computer resources (memory, CPU, I/O devices, files, etc.) to processes (user commands, application software, middleware services, database managers, other operating systems). An operating system also gives the users a command language to invoke the operating system facilities and to access the computing resources. This command language, also called control language in some systems, is used to access editors, compilers, utilities and other operating system resources.

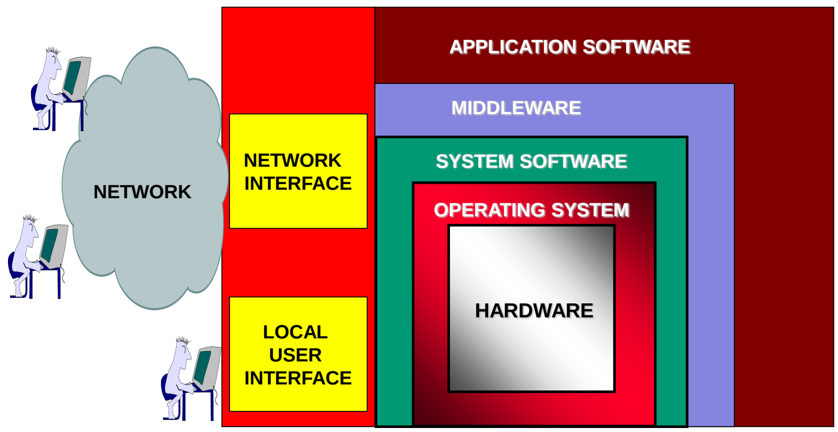


Figure 7-7: Interrelationships between Various IT Components

Many operating systems have been developed over the last thirty years. Examples of state of the market operating systems are Windows and its variants, Unix, Linux, AppleOS, and MVS. Typical functions performed by an operating system are:

- **Handle User Commands**. The operating system receives, parses and interprets the user commands. It also displays results of the user session with information about resource utilization, etc.
- **Allocation and Assignment.** The operating system allocates resources needed to execute the user commands and jobs, if the user is allowed to access the resources. It provides locations in primary memory for data and programs and controls the input and output devices such as printers, terminals, and telecommunication links.
- **Scheduling.** The operating system decides when to schedule the jobs and user requests that have been submitted. It also coordinates the scheduling in various areas of the computer so that as many things can be done in parallel as possible. The operating system must schedule jobs according to organizational priorities. On-line order processing may have priority over a job to generate mailing labels.
- **Monitoring.** The operating system monitors the activities of the computer system and the processes in the system. It keeps track of each computer job and may also keep track of who is using the system, of what programs have been run, and generate any error messages.
- **Security.** The operating system authenticates the user request for proper authority and keeps track of any unauthorized attempts to access the system.

Figure 7-8 shows a functional model of operating systems. The user command manager parses and executes the user commands. The memory manager allocates the main memory and the hardware registers to various processes. Most memory managers include the capabilities to manage virtual memory and to translate virtual to real memory. The CPU (central processing unit) manager allocates the CPU to the processes. A variety of CPU scheduling schemes (time slicing, interrupt driven) have been employed in the operating systems. The file managers manage the data resources in the system. This normally includes catalogs, file sharing, etc. The I/O (input/output) managers steward all the local and remote I/O activities. The network communication facilities may be included in some operating systems as I/O facilities.

A large number of operating systems are in commercial use at present. Examples are the Microsoft's Windows family of OSs (e.g., Windows 2007, Windows Millennium Edition, Windows NT, Windows Vista, Windows XP, and Windows CE), Mac OS (the operating system for the Apple computers), Unix family of operating systems, Linux (a Unix-like open source operating system that runs on Intel, Motorola, Mips, and other processors), and IBM's mainframe OSs (e.g., System Z OS). In addition, several specialized OSs also keep emerging (see the sidebar "Example of an Operating System: EeyeOS").

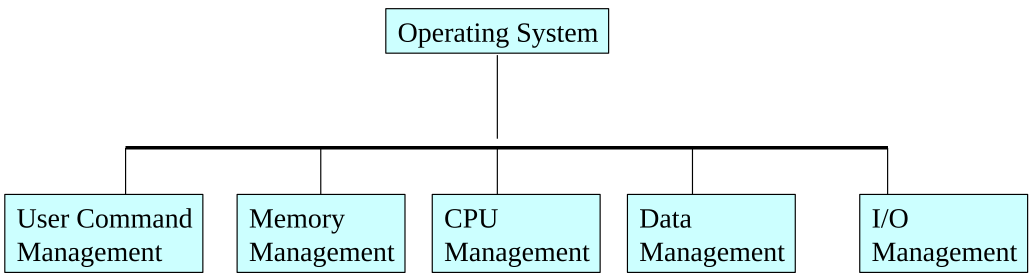


Figure 7-8: Centralized Operating System - Functional View

Example of an Interesting Operating System: EyeOS

EyeOS is a new kind of operating system, where everything resides on a web browser. It is an open source web desktop written mainly in PHP, XML, and JavaScript. EyeOS acts as a platform for web applications written using the eyeOS Toolkit. It includes a desktop environment with more than 60 applications and system utilities. It is accessible by portable devices via its mobile front end. Thus, with eyeOS, a user can have access to a desktop, applications and files from anywhere. A user has to just open a web browser, connect to her eyeOS System and access her personal desktop and all other resources as they were left when last accessed.

The OS is based on a client-server architecture, in which eyeOS is the server, and the client is usually a web browser. eyeOS has many services for specific tasks that are managed by the kernel. For developers, eyeOS provides the eyeOS Toolkit, a set of libraries and functions to develop applications for eyeOS. Using the integrated Portage-based eyeSoft system, one can create their own repository for eyeOS and distribute applications through it. EyeOS provides many advantages such as work from any place, easy sharing of resources, ease of use of files of different formats without any compatibility problems, and minimum problems due to computer crash or loss of data.

Sources:

- http://eyeos.org/en/?section=home
- http://en.wikipedia.org/wiki/EyeOS

7.3.1.1 Data and Transaction Managers

The data and transaction managers provide access and manipulation of user data and the operations performed on the data (see Figure 7-9). Examples of these important managers, discussed in more detail in Chapter 8, are:

- Database managers
- File managers
- Transaction managers

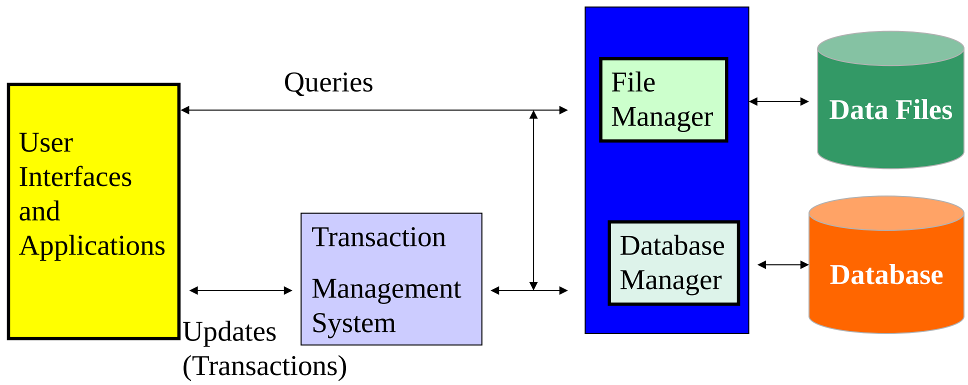


Figure 7-9: Conceptual View of Data and Transaction Managers

Database managers, also known as database management systems (DBMSs), provide access to databases for on-line and batch users. In a typical database environment, different users can view, access and manipulate the data in a database. A DBMS is designed to a) manage logical views of data so that different users can access and manipulate the data without having to know the physical representation of data, b) manage concurrent access to data by multiple users, enforcing logical isolation of transactions, c) enforce security to allow access to authorized users only, and provide integrity controls and backup/recovery of a database. Relational database managers such as DB2, Oracle, Sybase, and/or Informix are typically used in many contemporary applications. An important issue in modern environments is to access remotely located databases from a variety of customer devices.

File managers are responsible for providing access and manipulation of text documents, diagrams, charts, images, and indexed files. A very wide range of file managers have been developed since the 1960s. An important issue in distributed environments is to provide access to files that are dispersed around a network. This issue has been addressed by middleware such as the SUN Network File Server (NFS).

Transaction managers (TMs), also known as transaction processing monitor (TP monitor), monitor the execution of transactions (sequence of activities that must be executed as a unit). For example, online purchasing is a transaction that handles the payment, inventory and shipping activities. TMs specialize in managing transactions from their point of origin to their termination (planned or unplanned). Some TM facilities are integrated with the DBMS facilities to allow database queries from different transactions to access/update one or several data items (IBM's IMS DB/DC is an example). However, some TMs specialize in handling transactions only (CICS, Tuxedo and Encina are examples). The key issue in distributed environments is how to extend the scope of local transaction management to managing the execution of transactions across multiple sites.

7.3.1.2 Local Systems Software

The local software in a distributed environment provides access and manipulation of local data and processes located on networked machines. Examples of the local software are:

- Language translators
- Utility programs
- Print managers

Language translators (compilers) translate high-level language programs written in programming languages such as Java, C, C++, COBOL, FORTRAN, or C# into machine language that the computer can execute. The program in the high-level language before translation into machine language is called source code. A compiler translates source code into machine code called object code. Just before execution by the computer, the object code modules are joined with other object code modules in a process called linkage editing. The resulting load module is what is actually executed by the computer. Figure 7-10 illustrates the language translation process. Some programming languages such as BASIC do not use a compiler but an interpreter, which translates each source code statement one at a time into machine code during execution and executes it. Interpreter languages are very slow to execute because they are translated one statement at a time. Language translators also include assemblers that are similar to compilers, but are used to translate only assembly language (low level programs) into machine code.

Utility Programs. Local system software includes utility programs for routine, repetitive tasks, such as copying, clearing primary storage, checking for viruses, calendaring, calculators, clocks, or sorting. These utilities perform such functions as setting up new files, deleting old files, or formatting diskettes. Utility programs are prewritten programs that are stored so that they can be shared by all users of a computer system and can be used rapidly in many different information system applications when requested. Examples of utility programs are Norton Utilities, and Microsoft Accessories that consist of Notepad, Clock, and a Calculator, among other things.

Print managers are responsible for printing operations. Obviously, different types of print managers are available for different types of print devices. Almost all LANs at present provide access to print managers on the LAN.

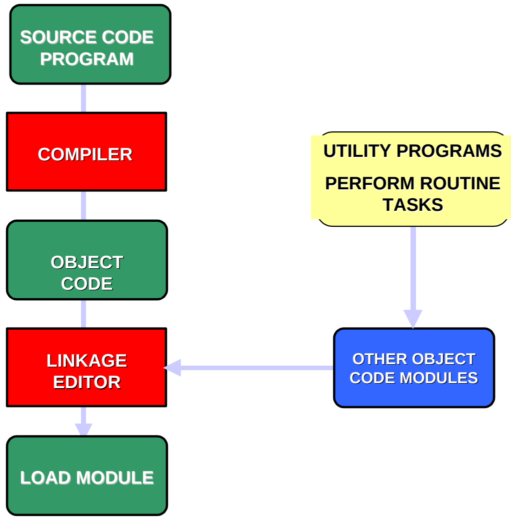


Figure 7-10: Language Translation Process

7.3.2 Web Technologies and the Internet at a Glance

Technically speaking, the world wide web (or just the Web), is a collection of the following technologies that operate on top of the Internet (see Figure 7-11):

- ***Web sites*** provide the content accessed by Web users. Web sites are populated and in many cases managed by the content providers.
- ***Web browsers*** are the clients that typically use graphical user interfaces to wander through the Web sites.
- ***Uniform Resource Locator (URL)*** is the basis for locating resources in WWW. A URL consists of a string of characters that uniquely identifies a resource.
- ***Hypertext Markup Language (HTML)*** is an easy-to-use language that tags the text files for display at Web browsers. HTML also helps in creation of hypertext links, usually called hyperlinks, which provide a path from one document to another.
- ***Hypertext Transfer Protocol (HTTP***) is an application-level protocol designed for Web users. It is intended for collaborative, distributed, hypermedia information systems
- ***Web navigation and search services*** are used to search and surf the vast resources available over the "cyberspace".
- ***Gateways to non-Web resources*** are used to bridge the gap between Web browsers and the corporate applications and databases. Web gateways are used for accessing information from heterogeneous data sources (e.g., relational databases, indexed files and legacy information sources) and can be used to handle almost anything that is not designed with an HTML interface.

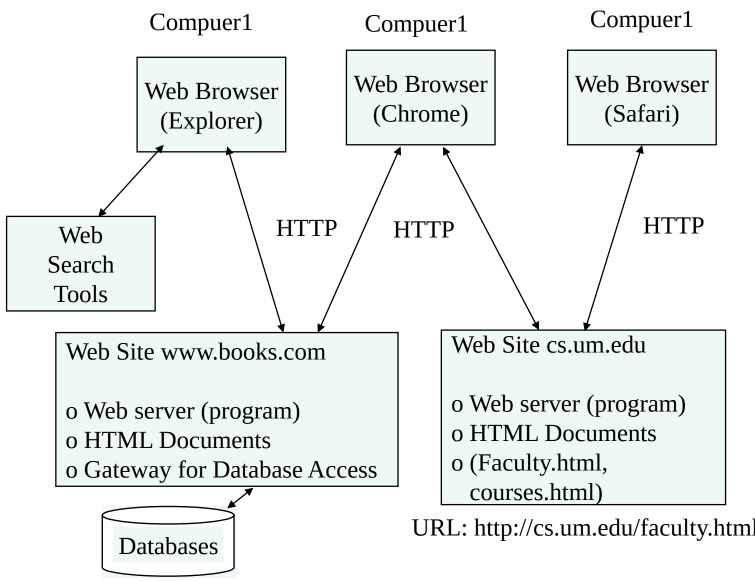


Figure 7-11: Conceptual View of World Wide Web

Network services provide a transport mechanism that is at the bottom of the IT infrastructure. Network services are provided through communication networks that are a collection of equipment and physical media, viewed as one autonomous whole, that interconnect two or more stations. A station is an end-point (source/sink) in a communication network and can be a terminal, computer, telephone, sensor or a TV. A network can be configured as a wide area network (WAN) for long distance communications, a local area network (LAN) for short-haul communications within a building, or a metropolitan area network (MAN) for communications within a city. Most enterprise networks are a combination of LANs, MANs, and WANs. Physically speaking, the communication between stations on a network can use analog or digital transmission facilities over copper, wireless or fiber optic media.

Wireless networks, as the name implies, interconnect devices without using wires -- instead they use the air as the main transmission medium. Wireless networks are enjoying widespread public approval with a rapidly increasing demand. The increase in the number of cellular phones, palm pilots, laptops, notebooks, and other handheld devices is phenomenal. To meet this demand, mobile communications technologies are emerging with digital speech transmission and the ability to integrate cordless systems into other networks. In the meantime, researchers are developing the next generation of technologies for several years to come.

The unique features of the wireless networks are:

- The bandwidths, and consequently data rates, of communication channels are restricted by government regulations. The government policies allow only a few frequency ranges for wireless communications.
- The communication channel between senders/receivers is often impaired by noise, interference and weather fluctuations.
- The senders and receivers of information are not physically connected to a network. Thus the location of a sender/receiver is unknown prior to start of communication and can change during the conversation.

Technically speaking***, Internet*** is a network of networks that is supported by the Internet Protocol (IP). What does this mean? Basically it says that you need to have an IP network (or a gateway that translates to IP) to join the Internet. Once you have an IP network, then you can run almost any physical network under it and take advantage of voice, data, or video applications for your e-business that run on top of IP. At present, the term Internet is used to symbolize a Public Internet that is not owned by any single entity -- it consists of many independent IP networks that are tied together loosely. The public Internet at present consists of millions of computers (PCs, Macs, Sun workstations, HP systems, IBM mainframes) that are interconnected through thousands of networks that use different underlying network technologies in different parts of the world. All these computers and networks are tied together through a global IP network via several wide area networks (WANs) (see Figure 7-12).

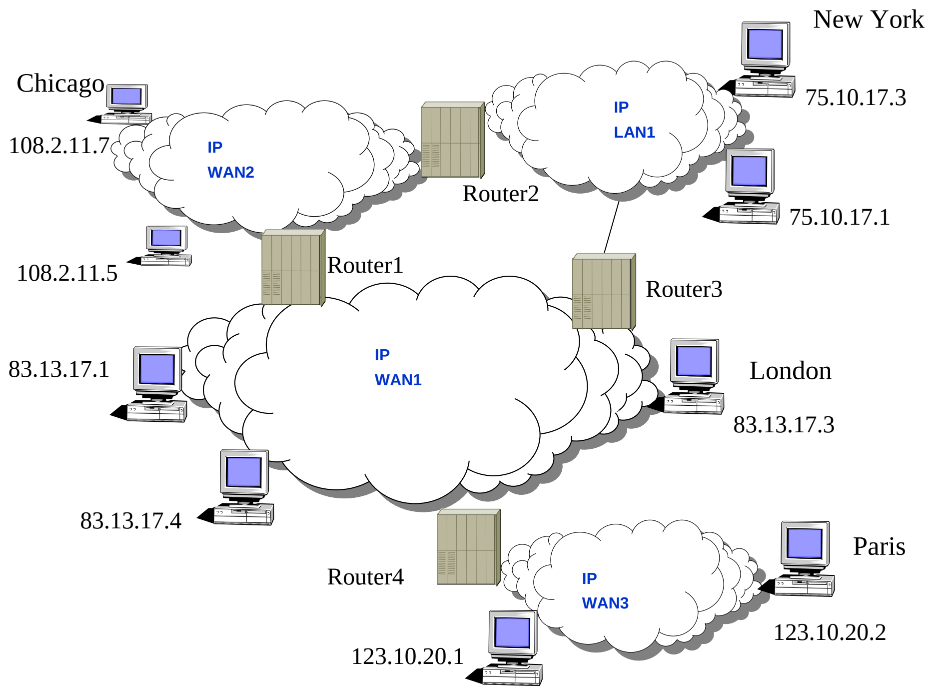


Figure 7-12: Public Internet – A Global IP Network

Suggested Review Questions Before Proceeding

- What is an operating system and what functions does it perform. List names of two operating systems.
- What is the difference between a database and transaction management system
- What is local system software and what does it do? List some examples of local system software that you are familiar with.
- What is the difference between a network and the Internet
- How do Web technologies utilize the Internet

7.4 IT Infrastructure Through Cloud Computing

7.4.1 Overview

Simply stated, cloud computing relies on sharing computing resources available over the Internet ("cloud") rather than having local computing or human resources. For example, a company can "rent" needed disk space, servers, application packages as well as technical staff to run its business instead of buying everything and hiring staff. The main idea is that companies can quickly and inexpensively rent tremendous resources available on the cloud – the Internet. The two major players in cloud computing are (see Figure 7-13):

- Cloud Providers that offer a set of computing, applications, and other resources that can be rented, for a fee
- Cloud Consumers, a large number of companies, that rent the needed services from the cloud providers instead of individually owning the needed hardware/software

Basically, cloud computing offers shared IT infrastructure that contains large pools of systems that are linked together. In the beginning cloud services were mainly cheap devices and applications but with time very high scale and sophisticated services are being provided over the cloud. A good example is the IBM CAMS (Cloud, Analytics, Mobile, and Security) services that are offered over the cloud for Big Data applications.

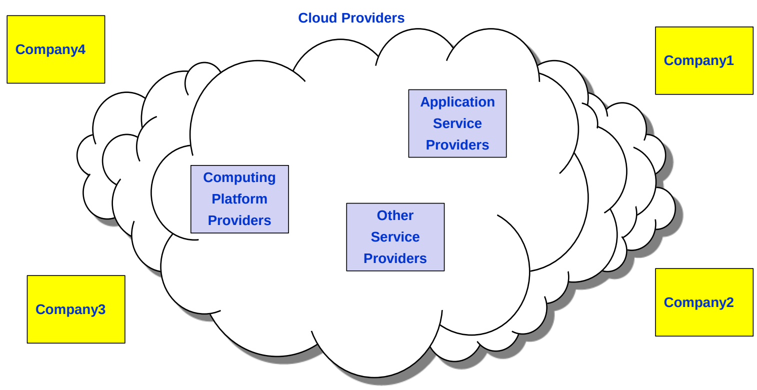


Figure 7-13: Conceptual View of Cloud Computing

For example, a cloud service provider offers a set of services based on an agreed upon contract. The services can range from disk space in the cloud to Web hosting and to complete IT infrastructure hosting where *all* IT services are in the cloud – the companies just have terminals that connect to the cloud. Cloud computing allows companies to outsource a few or all of its infrastructure services.

Cloud computing is fueled by the benefits in cost reduction, increased flexibility, and the speed of implementation. However, it introduces risks in privacy, security and loss of control. We will discuss the benefits and the risks of cloud computing in more detail later. To fit the needs of different users, a cloud can be deployed as a public, private or a hybrid cloud. Specifically:

- Public cloud is a shared resource offered by service providers. As the name suggests, public cloud enables many clients to share resources on a cloud and the resources are offered to the clients on an as-needed basis. Public cloud is accessible from the internet, is externally hosted, and used by the general public. It is cheaper but raises security and privacy issues.

- Private cloud is owned by an enterprise and is dedicated to its needs. It can grow as the organization grows. Private cloud is accessible from an intranet, is internally hosted, and used by a single organization.
- Community cloud has the resources and the infrastructure accessible to a specific community such as a city, a state government or a consortium. It is basically an extranet that is available only to the members of the community.
- Hybrid cloud is a combination of two or more clouds.

The most common configuration at present is the public cloud, shown in Figure 7-14.

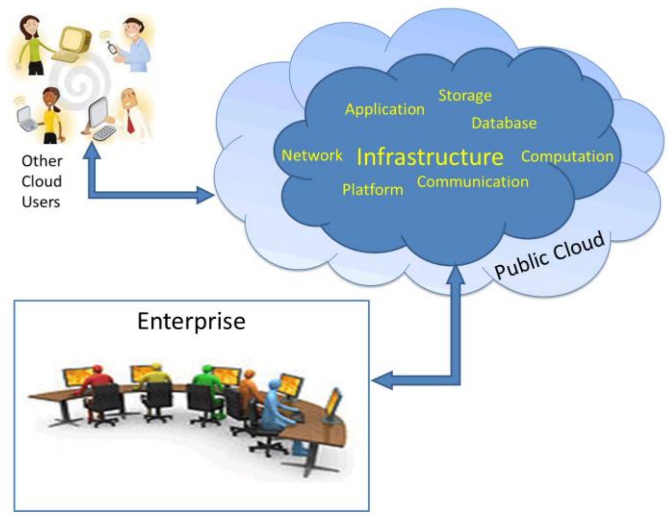


Figure 7-14: A Typical Public Cloud Computing Environment

7.4.2 Cloud Service Models and Service Providers

In essence, anything and everything can be outsourced to the cloud. Different service provider models, centered around the Cloud, are available at present. For example, businesses and consumers can rent services from cloud service providers such as the following to support different service models (see Figure 7-19):

- *Infrastructure as a Service (IaaS)* provides access to server machines, storage, network transport, and other fundamental computing resources. This service model is supported by *Network Service Providers (NSPs)* that provide the network "pipe" (end to end network communication and routing services) for digital businesses. Examples of NSPs are the telecommunications companies that include a variety of local exchange carriers and long distance carriers. *Internet Service Providers (ISPs)* also support IaaS by providing access to the public Internet and web services. Verizon and Comcast are well known examples of ISPs.
- *Platform as a service (PaaS)* provides access to the platform services (operating systems, disk space, basic middleware) needed to support business applications for different industry sectors. Platform Service Providers (PSPs), in essence, are similar to the old "computing centers" that provided the computing hardware/software for business applications. For example, Amazon.com provides PaaS for disk space on the cloud,
- *Software as a service (SaaS)* provides access to a provider's software application packages. Many *Application Service Providers (ASPs)* host application packages for different industry sectors that clients access over the Internet for a fee. A very wide range of ASPs have emerged in recent years with services that range from payroll to large scale ERP systems. For example, major

software vendors such as Microsoft, IBM, SAP, Oracle, and Peoplesoft have become ASPs. We will discuss ASPs in more detail later.

In addition, several *Business Service Providers (BSPs)* provide business services such as mail delivery, customer support, and building security. Several variations of these service providers can be also envisioned in different industry sectors such as health, education, public safety, public welfare, transportation, finance and agriculture. .

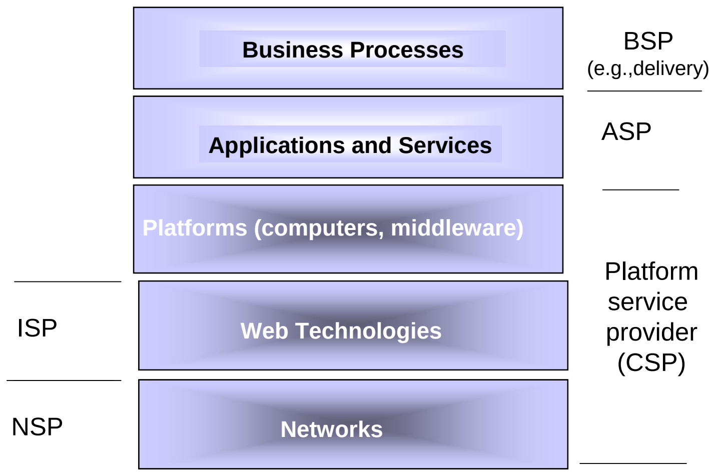


Figure 7-15: Types of Service Providers

Examples of Cloud Services

Few examples of major corporations providing cloud services:

- Amazon (amazon.com) offers disk space through Amazon Web Services (AWS), besides selling books and other items on the Internet.
- Cisco (cisco.com) offers extensive networking and data center support services over the cloud.
- Google (google.com) offers many services over the cloud such as Gmail and Google Docs.
- IBM (ibm.com) offers CAMS (Cloud, Analytics, Mobile, and Security) services that are offered over the cloud for Big Data applications.
- SAP (sap.com) offers a large number of its SAP ERP systems over the Internet.

The following site lists Top 150 Cloud Providers : http://cloudcomputing.sys-con.com/node/770174
In addition, the following types of cloud services are available over the Internet:

- Social Networking in the cloud are offered by Facebook, LinkedIn, MySpace, Twitter, and many, many others.
- Data Backup Services are offered by providers such as Dropbox
- Legal Services over the cloud are provided by legal zoom.com.
- Banking and Financial Services are being offered by most large banks such as Citicorp and WellsFargo.
- Health Care services are being provided over the cloud by the US Department of Health and Human Services' (HHS) and other healthcare providers.
- Mobile apps for Android and iPhone handsets are provided over the cloud by a very large number of mobile app developers.

Additional examples of different types of Cloud Computing Services can be found at http://epic.org/privacy/cloudcomputing/. .

7.4.3 Benefits and Risks in Outsourcing to the Cloud

Outsourcing, i.e., hiring someone else to do a job on your behalf, has been an attractive business practice for several years. Service providers (SPs) in the cloud (e.g., NSPs, ISPs, PSPs, and ASPs) are making it possible for companies to outsource many of their services. It is theoretically possible for a newly formed company to outsource *everything* by using a variety of cloud service providers. That is the essence of ***virtual enterprises*** – they rely exclusively on outsourcing by using cloud computing where everything is outsourced to the Internet cloud. In particular, the role of application service providers (ASPs) is of particular importance to modern business outsourcing. A large number of ASPs are currently available over the cloud.

Cost and time savings are the two basic drivers for outsourcing through SPs. Figure 7-16 shows the basic cost motivation. Let us take the example of a start-up software company to illustrate the basic idea. This company needs to use a variety of applications such as payroll, inventory, purchasing, order processing, etc. to get started. Buying, installing, and running these applications and the underlying platforms would require an up-front cost even when there are very few customers (or none). Outsourcing, i.e., renting these applications from an SP can be much more economical on a per usage basis. However, as the customer base grows, the cost of renting may exceed the cost of owning.

In addition to cost savings, time savings can be a big factor in outsourcing – you can use services more quickly from a service provider than by struggling through your own purchase, acquisition, and install cycle. According to the ASP Forum (www.aspforum.com), building your own infrastructure requires a minimum of 8 months to a year for a small company while it can be done in a matter of weeks by using SPs. For example, ASPs like Corio can support a new customer in 4 to 14 weeks for large applications that may require several months to purchase and set up. More information about different aspects of outsourcing can be found at the site (www.outsourcing-research.com).

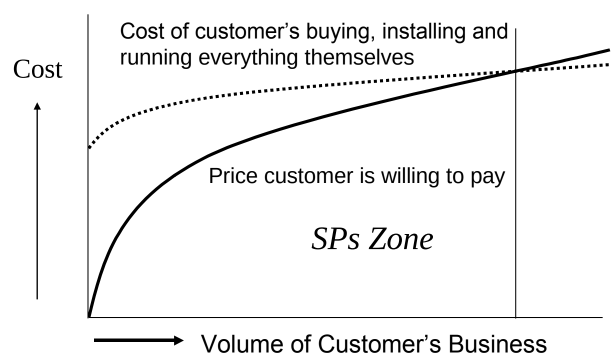


Figure 7-16: Cost Motivation for using Service Providers

Besides cost and time savings, flexibility and scalability are additional benefits of cloud computing. However, it introduces risks in privacy, security and loss of control. Table 7-1 lists in detail some of the key benefits and risks of cloud computing. This table can be easily extended.

Table 7-1: Benefits and Risks of Cloud Computing

Benefits of Cloud Computing	Risks of Cloud Computing
•Lower Capital and Operating Costs: Organizations only pay for the actual capacity used by just renting the needed computing resources from cloud service providers. If needed, organizations can rent added resources (e.g., disk or server space) for a few hours •Focus on Critical Business Functions: By outsourcing support services to the cloud, organizations can focus their attention on mission critical business services. •Flexibility in Selection of Optimal IT Infrastructure: Organizations can quickly access new technologies without high risk and cost. The organizations can also experiment with new things with the ability to revert back to their original system if things do not work out. •Increased Versatility and Scalability: The pay-per-use model used in cloud computing makes it versatile for large corporations and affordable for small ones. Basically, cloud providers can offer an infinite amount of resources to a large number of users. •Increased Reliability: Cloud vendors have to provide 24/7 technical support and highly trained experienced staff to support the infrastructure to stay in business. Small organizations cannot afford to provide the same high level of support. •Possible Security Gains for Small Companies: Although there are security risks with outsourcing your data assets, many cloud vendors can offer better security services to small companies with limited security skills.	•Data Privacy and Security Issues: Moving confidential data to cloud service providers is a major risk for most organizations. Offsite Data at remote locations has to travel more and is thus easier to be intercepted. Security technologies are improving but so are the skills of the hackers. •Centralization Concerns: Cloud owners are good targets of physical and technical attacks due to their size and significance. The seizure of data-hosting centers by someone else is a major concern. •Record Retention and Disaster Recovery Requirements: If your business is subject to special record retention and data availability requirements, then you must make sure that the cloud provider is fully aware of these requirements especially in disaster situations. •Internet Dependency: You need Internet connection to access your cloud resources. Thus your work is totally dependent on the availability of Internet connection free of service interruptions at any time. •Difficulties with Legacy Systems: Many public and private organizations heavily rely on legacy systems for their day-to-day operations. These legacy systems are key players that cannot be easily outsourced to the cloud because most cloud owners cater to newer applications and technologies. This creates complex systems where some services of an organization are in the cloud but the legacy systems are not. •Cloud Provider Selection Issues: There are many cloud providers with different pricing and service level agreements. It is difficult to select the right cloud provider that also produces third-party audit reports of their capabilities.

7.4.4 Tradeoff Analysis and Service Level Agreements

The benefits and risks associated with cloud computing, as shown in Table 7-1, can serve as the key factors that drive the cloud computing decisions.

- Security and privacy requirements (minimal, medium, high)
- Ownership and control requirements (low, medium, high)

- Cost considerations (low, medium, large)
- In-house expertise available for developing and managing systems (low, medium, high) ,
- Usage requirements, e.g., number of transactions
- Flexibility requirements, e.g., changes per month (low, medium, high)
- Scalability Requirement Low (system not likely to grow)
- Availability Requirement Low (30% downtime)

These factors can be applied to select cloud services for networks, platforms, application software, IT staff, and other resources (see Table 7-2). In general, security, privacy, ownership and economic considerations drive the decisions of private versus public cloud. Most companies use the following hybrid cloud approaches:

- Use public cloud as much as possible for the resources that can benefit most from the public cloud
- Use private cloud for the type of data and other services that cannot be moved to the public cloud

Companies can start by outsourcing a few services to the public cloud and then move more to the cloud based on the lessons learned. Cloud solutions should be considered carefully for developing countries, especially considering low availability of Internet connectivity.

An important step in using cloud computing is establishing a Service Level Agreement (SLA) that minimizes the risks of cloud computing. Sidebar "Service Level Agreement" shows a template of a typical SLA.

Table 7-2: A Simple Tradeoff Analysis for Cloud Computing

Key Factors	**Computing Platforms**	**Network Services**	**Application Software**	**Staff**
Security and privacy requirements				
Ownership and control requirements				
Cost considerations				
In-house expertise available for developing and managing systems				
Usage requirements				
Flexibility requirements				
Scalability Requirement				
Availability Requirement				

Service Level Agreements(SLA)

A Service Level Agreement (SLA) is an essential aspect of cloud computing or any type of outsourcing because it mitigates risks for outsourcing. An SLA typically specifies the following:

- Names of both parties, Service provider and Service consumer
- What Resources are offered
- Service Availability and uptime and procedures to take care of failures
- What kind of maintenance is available
- Where the data and applications will reside
- What type of disaster recovery and business continuity planning will be available
- What assurances will be provided about the confidentiality of the data being outsourced
- What type of processes will be used to discontinue a service
- What will be the QoS in terms of performance and security for a particular service

Recommended Sources:

1. http://www.ibm.com/developerworks/cloud/library/cl-rev2sla.html?ca=drs-
2. http://www.sla-zone.co.uk/
3. http://viodi.com/2011/01/13/what-should-cloud-computing-users-and-providers-consider-for-slas/
4.http://www.jisclegal.ac.uk/ManageContent/ViewDetail/ID/2141/User-Guide-Cloud-Computing-Contracts-SLAs-and-Terms-Conditions-of-Use-31082011.aspx

7.4.5 Sample Architectures for Cloud Computing

The private, public, and community clouds can be used to architect different scenarios for modern enterprises. Let us go through some samples.

Applications in the cloud using ASPs (Application Service Providers) are playing a key role in digital enterprises. As stated previously, an ASP (Application Service Provider) manages and delivers application capabilities to multiple entities from a data center in the cloud. A wide range of ASPs market and deliver standardized application software over the Internet through a usage-based or transaction-based pricing contract.

Figure 7-17 shows a conceptual view of a large scale ASP that offers a wide range of business applications in the public cloud. A very large number of ASPs such as SAP, Oracle, Microsoft and others use this architecture. Enterprises can rely exclusively on one or more ASPs in the public cloud to run their businesses or offer some applications locally in the private cloud as shown in Figure 7-17.
.

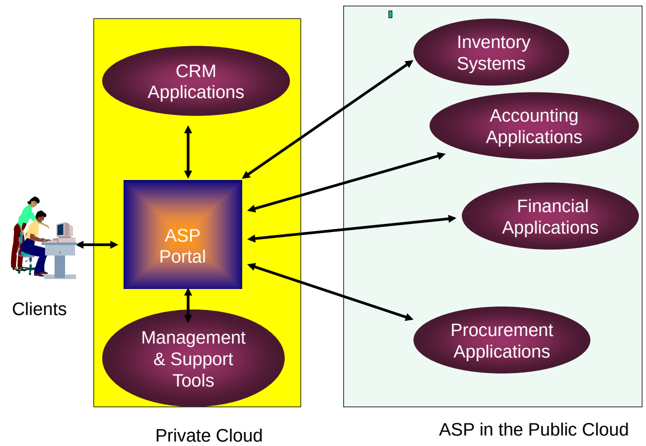


Figure 7-17: Conceptual View of an ASP

The ASP business model, as mentioned previously, is service and subscription-based. The clients of an ASP pay a fee, depending on the type of application and number of users, to use the application over the Internet. For example, a user that needs to use an SAP application for a few months but does not want to purchase SAP, can “rent” the use of SAP from an application service provider. It is also possible for a user to rent applications from different ASPs depending on the application needs and the ASP fees. Different ASPs have different fee structures. For example, some ASPs only cater to large customers that need support for more than 100 users for more than a year, while others provide a daily and weekly rate. The types of applications also vary widely. At present, you can find an ASP for almost any type of application – including desktop applications (e.g., word processing), games, and serious business applications such as payroll, asset management, and inventory control. Most major software vendors such as SAP, Oracle, and Peoplesoft have become ASPs over the cloud.

The ASP industry is growing in "Internet time” because this model appeals to many IT managers. However, due to a plethora of ASPs, it is impossible for IT decision makers to choose. It is difficult to keep up with the different offerings, value propositions, and business policies of so many firms in order to select the very best ASP to meet their needs. ASP emphasizes server side software, i.e., complex software such as application servers, workflow engines, enterprise application integrators, and message brokers all reside at the server sites. The clients are relatively light weight, i.e., they use a minimum of application specific software on the client side (“thin clients”). This thin client model allows even the smallest companies to take advantage of the ASP marketplace.

Figure 7-18 shows a more detailed view of the ASP model with more emphasis on the ASP Portal. Many ASPs at present offer a “portal” type access to the available applications (hence the term “ASP portal”). The customers access these applications through the ASP front-end based on security and access rules. The ASPs may house the applications and may also directly connect with remote applications through an integration layer (i.e., become an ASP gateway). According to this architecture, the front-end of an ASP Portal must be able to handle multitude of devices while the back-end must be able to communicate with multiple applications. The middle tier provides the security and access control for the users.

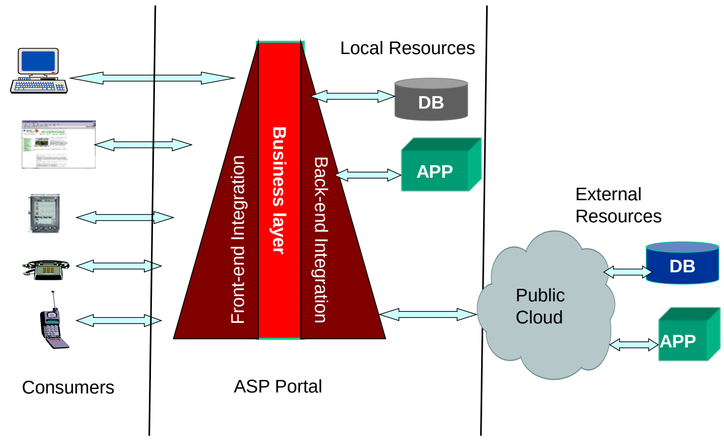


Figure 7-18: Detailed View of an ASP Portal for Cloud Services

Total Cost of Ownership

Total cost of ownership (TCO) determines the total economic value of an investment and includes the direct and indirect costs of a product or system. Direct cost is the cost of acquiring a system and the indirect cost includes operating plus replacement or upgrade costs.

In IT industries, the direct costs typically include the following:

- Computer hardware and programs
- Network hardware and software
- Installation and integration costs of hardware and software
- Warranties and licenses
- License tracking - compliance
- Migration expenses from one site to another

The indirect costs typically include the following:

- Operational expenses such as the IT personnel, management time, floor space, electricity, testing costs, physical security, backup and recovery processes, and training costs.
- Long term expenses such as replacements, upgrade, documentation and maintenance expenses

For IT, the direct cost is only about 30% of the TCO, about 70% is the indirect cost. The main advantage of cloud computing is that it significantly reduces the indirect costs.

7.5 Short Case Studies and Analysis

7.5.1 Discussion: Will Cloud Computing Eliminate the Need for IT Infrastructure

Cloud computing (CC) is essentially an outgrowth of the service provider (SP) model where an SP offers a set of services based on an agreed upon contract. In a CC model, all IT-related capabilities are provided "as a service" (e.g., software as a service (SaaS), platform as a service (PaaS), Infrastructure as a service (Iaas)), etc. Different providers, residing in the cloud (e.g., the Internet) can provide these services and are available to the users (see Figure 7-19). This allows the users to access a very wide range of technology-enabled services from the Internet without ownership, control, knowledge, or technical expertise of the complex IT infrastructure. The main appeal of the CC model is that the companies do not need to own and operate complex IT systems – they can just run them.

Naturally, this raises the question: do the companies need to have any IT infrastructure at all besides a browser? Are there some negative aspects of CC? Yes, of course. CC raises several issues such as the following: what are the security and privacy vulnerabilities, who is responsible for quality of service, what happens if a cloud vendor goes out of business, how will the applications between the vendors be integrated with each other, etc. Basically, the companies have to understand the issues before completely outsourcing everything.

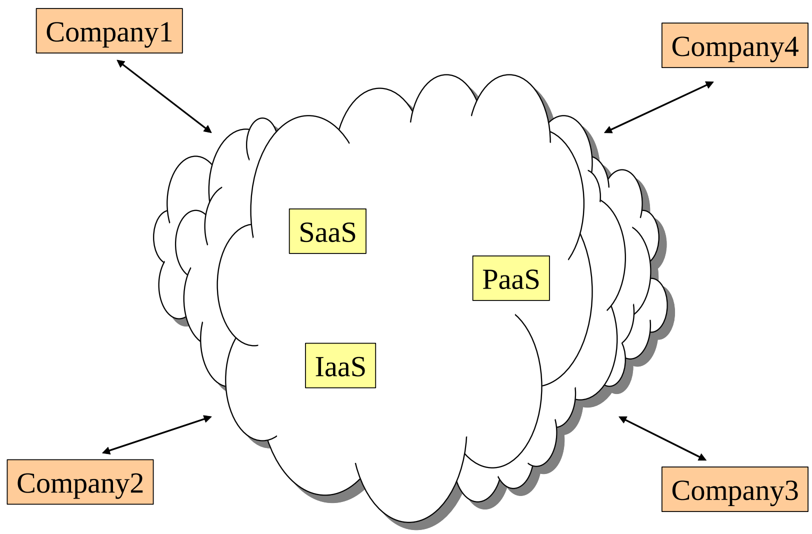


Figure 7-19: Conceptual View of Cloud Computing

7.5.2 Cloud Case Study: Sales force.Com Model

Salesforce.com's CRM applications use *Software as a Service* (SaaS) and *Platform as a Service* (PaaS). SaaS is a software application delivery model where a software vendor develops a web-native software application and hosts and operates the application for use by its customers over the internet. Customers do not pay for owning the software - rather they pay for using it. They use it through an [application programming interface] API accessible over the web.[1]

Platform as a Service (PaaS) is an outgrowth of the SaaS application delivery model. The PaaS model makes all of the facilities required to support the end-to-end life cycle of building and delivering web applications and services entirely available from the Internet[2]—with no software downloads or installation for developers, IT managers or end-users.

PaaS offerings include workflow facilities for application design, application development, testing, deployment and hosting as well as application services such as team collaboration, web service

[1] http://en.wikipedia.org/wiki/Software_as_a_Service
[2] http://blogs.zdnet.com/Hinchcliffe/?p=166&tag=btxcsim

integration and marshalling, database integration, security, scalability, storage, persistence, state management, application versioning, application instrumentation and developer community facilitation. These services are provisioned as an integrated solution over the web.[3]

Source: www.salesforce.com

7.5.3 Anatomy of an IT Disaster: How the FBI Blew it

This case study is essentially about the FBI's attempt to overhaul and modernize its aging technology infrastructure in the areas of enterprise-wide upgrade of desktop hardware and software; deployment of a modern network infrastructure; and an integrated suite of software for entering, finding, sharing, and analyzing case information.. The project to accomplish this was called Trilogy and soon became an icon of gigantic failure. The project became known infamously as the Tragedy project and is a classic example of catastrophic miscommunication and staggering waste. Originally estimated to cost $170 million it ballooned to $581 million. It burned through five different CIOs, 10 project managers, and 36 contract changes.

Trilogy encompassed the deployment of 30,000 computers, 4,000 printers, 1,600 scanners, 465 servers, and 1,400 routers as of April 2004. Trilogy's has had little impact on the FBI's antiquated case-management system, which today remains a myriad of mainframe green screens and vast stores of paper records. The software known as VCF (Virtual Case File) is the problem- it isn't in production yet and may never be. VCF is viewed to be the most extreme case of requirements bloat in IT history. As a result the Trilogy failure leaves the FBI with its obsolete, mainframe based ACS (Automated Case Support) system, which requires the user to traverse a dozen 3270 green screens to upload a single document.

A detailed autopsy of the Trilogy project is beyond the scope of this assignment but the salient points of it that contributed to the majority of the project's failure point to seven distinct things:

- Lack of requirements and failure to lock them down
- Constant scope creep
- Using a contract payment structure (cost-plus-award) that incents the contractor to accept out of scope changes *(A performance based contract should have been used instead.)*
- Poor communications with all levels of stakeholders. Didn't manage expectations very well.
- A contractor can't modernize a client's technology infrastructure that is without an extensible enterprise architecture.
- Having flash cutover as a requirement
- Client's mission and operational needs are in a constant state of flux.

Sources:

- Knort, E., "Anatomy of an IT Disaster: How the FBI blew it", Information Week,, March 21, 2005, Link: http://www.infoworld.com/print/15243
- Eggen,D. and White, G., "The FBI Upgrade That Wasn't -- $170 Million Bought an Unusable Computer System", Washington Post, August 18, 2006; A01, Link: http://www.washingtonpost.com/wpdyn/content/article/2006/08/17/AR2006081701485_pf.html

[3] http://en.wikipedia.org/wiki/Platform_as_a_service#cite_note-0

7.6 Summary

An overall Enterprise Architecture that identifies the key building blocks of an enterprise and establishes the interrelationships between these building blocks is a good starting point for an enterprise planning project. The top two layers of an enterprise architecture concentrate on the business issues while the lower two layers deal with the enabling technology (IT infrastructure) considerations. IT infrastructure is the backbone of modern digital enterprises. We have reviewed the key concepts and building blocks of the IT infrastructure in this chapter. In summary:

- Computing platforms consist of operating systems and computing hardware to provide the basic scheduling and hardware services.
- Databases house the data for day-to-day operation plus support business intelligence and management decision making.
- Networks provide the network transport between remote parties and are responsible for routing and flow/error control support.
- Middleware interconnects remotely located users, databases and applications. Web browsers and servers are best known examples of middleware.

Cloud computing is an interesting trend that allows the users to rent everything (computers, networks, system software, etc) thus potentially eliminating the need for owning any IT infrastructure.

Cloud computing (CC) is essentially an outgrowth of the service provider (SP) model where an SP offers a set of services based on an agreed upon contract. In a CC model, all IT-related capabilities are provided "as a service" (e.g., software as a service (SaaS), platform as a service (PaaS), Infrastructure as a service (Iaas), etc. Different providers, residing in the cloud (e.g., the Internet) can provide these services and are available to the users (see Figure 7-19). This allows the users to access a very wide range of technology-enabled services from the Internet without ownership, control, knowledge, or technical expertise of the complex IT infrastructure. The main appeal of the CC model is that the companies do not need to own and operate complex IT systems – they can just run them.

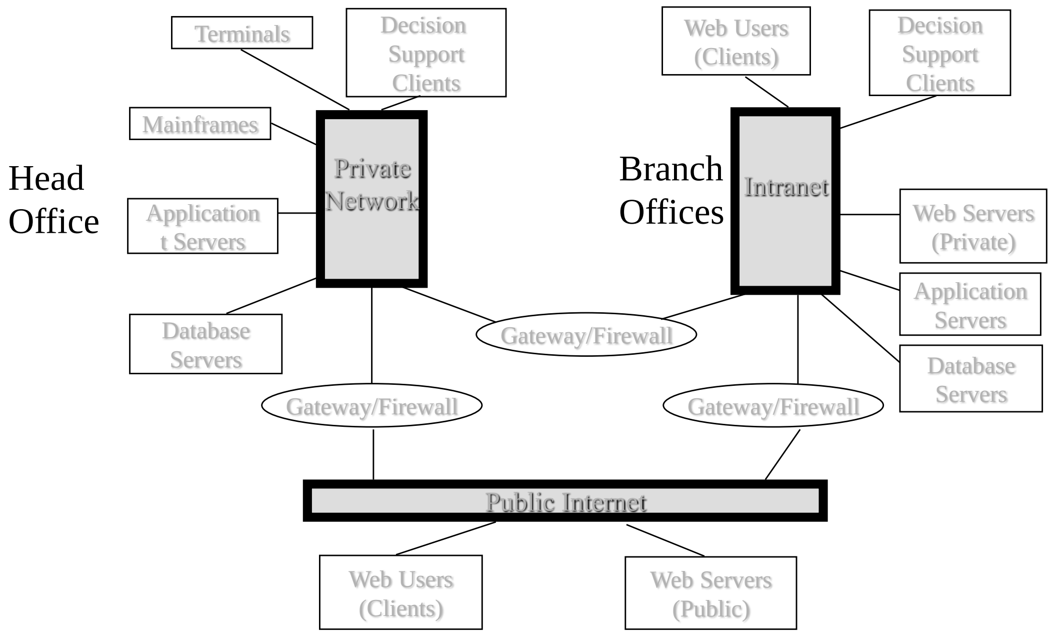


Figure 7-20: XYZCorp IT Infrastructure High Level View

7.7 Review Questions and Exercises

- What is enterprise architecture and what are its key components
- Present a different view of how client/server systems interrelate with distributed computing systems.
- Give an example of a distributed application that does not use the client/server model.
- Describe at least three different ways that you can invoke distributed object services from Web browsers, including applets.
- Draw a conceptual diagram of IT infrastructure in an environment of your choice (e.g., business, finance, engineering, manufacturing).
- List the factors you will use to evaluate off-the-shelf middleware to support modern business applications.
- Review the literature on cloud computing (CC) and evaluate how it could be used to run a small to medium business. What are the strengths and weaknesses of cloud computing.

7.8 Additional Information

[1] Allen, P., "Realizing e-business with Components", Addison Wesley Professional; 2000.
[2] Bell, T., Adam, J., and Lowe, S., "Communications", IEEE Spectrum 1996, Jan. 1996, pp. 30-41.
[3] Birrel, A.D. and Nelson, B.J., "Implementing Remote Procedure Call", ACM Trans. on Computer Systems, Vol. 2, pp.39-59.
[4] Booch, G., "Object Oriented Design with Applications", Benjamins Cummings, Second edition, 1994.
[5] Comerford, R., "Computers", IEEE Software, Jan. 1996, pp. 42-25.
[6] Comport, J., "Packaged Applications: Buy an Application, Inherit an Architecture", Gartner Group Briefing, San Diego, February 1995.

[7] Conard, J., et al, "Introducing .NET", Mass Market, 2001.
[8] Coulouris, G., and Dollimore, J., "Distributed Systems: Concepts and Design", Addison Wesley, 3rd edition, 2000.
[9] Davydov, M., "Corporate Portals and e-business Integration", McGraw-Hill Professional Publishing; 2001.
[10] Elsenpeter, R., and Velte, T., "e-business: A Beginner's Guide", McGraw-Hill Professional Publishing; 2000.
[11] IDC, "The e-business Platform: Where Application Servers and Application Integration Meet", 2001.
[12] Marcus, E., and Stern. H., "Blueprints for High Availability: Designing Resilient Distributed Systems", Wiley & Sons, 2000.
[13] Sawhney, M., et al, "The Seven Steps to Nirvana: Strategic Insights into e-business Transformation", McGraw-Hill Professional Publishing, 2001.
[14] Tannenbaum, A., "Modern Operating Systems", Prentice Hall, 1992.
[15] Taylor, D., "Object Oriented Technology: A Manager's Guide", Addison-Wesley, 1994.
[16] Trllica, C. "Software Applications", IEEE Spectrum, Jan. 1996, pp. 56-59.
[17] Umar, A., "Object/Oriented Client/Server Internet Environments: Enabling Distributed Enterprises Through Middleware", Prentice Hall, 1997.
[18] Umar, A., "Distributed Computing and Client/server Systems", Prentice Hall, revised edition, 1993.
[19] Wilbur, S. and Bacarisse, B., "Building Distributed Systems with Remote Procedure Calls", Software Engineering Journal, Sept. 1987, pp. 148 - 159.
[20] Whyte, W., and Whyte; B., "Enabling e-business - Integrating Technologies Architectures & Applications", John Wiley & Sons; 2001.
[21] Wooding, T., "Business Frameworks", Object Magazine, January 1997, pp. 50-53.
[22] Wu, J., "Distributed System Design", CRC Press; 1998.
[24] W. Kim, "Cloud Computing Architecture", International Journal of Web and Grid Services, 2013 -