**Microsoft’s Internet Information Server IIS**

* Although Apache has been ported to the Windows platforms, it is not the most popular server on those systems.
* Because the Microsoft **IIS** server is supplied as part of Windows
* And because it is a reasonably good server—most Windows-based Web servers use **IIS**.

Apache and IIS provide similar varieties of services. From the point of view of the site manager,

What the difference between **Apache** and **IIS**

* Apache is controlled by a configuration file that is edited by the manager to change Apache’s behavior.
* IIS, server behavior is modified by changes made through a window-based management program, named the **IIS snap**-in, this program allows the site manager to set parameters for the server.

**Uniform Resource Locators (URL)**

Uniform (or universal) resource locators (URLs) are used to identify documents (resources) on the Internet. There are many different kinds of resources, identified by different forms of URLs.

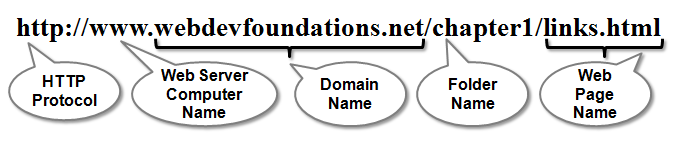
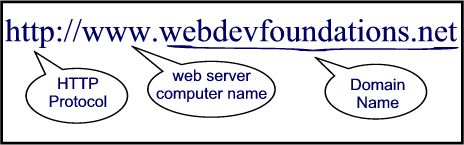
**URL Formats**

All URLs have the same general format: Scheme: object-address

* The scheme is often a communications protocol.
* Common schemes include http, ftp, gopher, telnet, file, mail, and news.
* Different schemes use object addresses that have different forms. Our interest here is in the HTTP protocol, which supports the Web.

This protocol is used to request and send extensible Hypertext Markup Language (XHTML) documents. In the case of HTTP, the form of the object address of a URL is as follows:

**//fully-qualified-domain-name/path-to-document**

****

URLs can never have embedded spaces. Also, there is a collection of special characters, including semicolons, colons, and ampersands (&), that cannot appear in a URL. To include a space or one of the disallowed special characters, the character must be coded as a percent sign (%) followed by the two digit hexadecimal ASCII code for the character. For example, if San Jose is a domain name, it must be typed as San%20Jose (20 is the hexadecimal ASCII code for a space).

**1.6 Multipurpose Internet Mail Extensions MIME**

A browser needs some way of determining the format of a document it receives from a Web server. Without knowing the form of the document, the browser would be unable to render it, because different document formats require different rendering tools. The forms of these documents are specified with Multipurpose Internet Mail Extensions (MIME).

**Type Specifications**

* MIME was developed to specify the format of different kinds of documents to be sent via Internet mail.
* These documents could contain various kinds of text, video data, or sound data. Because the Web has needs similar to those of Internet mail, MIME was adopted as the way to specify document types transmitted over the Web.
* A Web server attaches a MIME format specification to the beginning of the document that it is about to provide to a browser.
* When the browser receives the document from a Web server, it uses the included MIME format specification to determine what to do with the document. If the content is text, for example, the MIME code tells the browser that it is text and also indicates the particular kind of text it is. If the content is sound, the MIME code tells the browser that it is sound and

Then gives the particular representation of sound so that the browser can choose a program to which it has access to produce the transmitted sound. MIME specifications have the following form: type/subtype

* The most common MIME types are text, image, and video.
* The most common text subtypes are plain and html. Some common image subtypes are gif and jpeg.
* Some common video subtypes are mpeg and quick time.

A list of MIME specifications is stored in the configuration files of every Web server. In the remainder of these lectures, when we say document type, we mean both the document’s type and its subtype. Servers determine the type of a document by using the filename’s extension as the key into a table of types. For example, the extension .html tells the server that it should attach text/html to the document before sending it to the requesting browser. Browsers also maintain a conversion table for looking up the type of a document by its file name extension.

**Security**

On the Web server side, anyone on the planet with a computer, a browser, and an Internet connection can request the execution of software on any server computer. He or she can also access data and databases stored on the server computer. On the browser end, the problem is similar: Any server to which the browser points can download software that is to be executed on the browser host machine. Such software can access parts of the memory and memory devices attached to that machine that are not related to the needs of the original browser request. In effect, on both ends, it is like allowing any number of total strangers into your house and trying to prevent them from leaving anything in the house, taking anything from the house, or altering anything in the house. The larger and more complex the design of the house, the more difficult it will be to prevent any of those activities. The same is true for Web servers and browsers: The more complex they are, the more difficult it is to prevent security breaches. Today’s browsers and Web servers are indeed large and complex software systems, so security is a significant problem in Web applications. In this section one can give no more than a brief sketch of some of the subtopics of security. One aspect of Web security is the matter of getting one’s data from the browser to the server and having the server deliver data back to the browser without anyone or any device intercepting or corrupting those data along the way. Consider just the simplest case that of transmitting a credit card number to a company from which a purchase is being made. The security issues for this transaction are as follows:

1. **Privacy**—it must not be possible for the credit card number to be stolen on its way to the company’s server.

2. **Integrity**—it must not be possible for the credit card number to be modified on its way to the company’s server.

3. **Authentication**—it must be possible for both the purchaser and the seller to be certain of each other’s identity.

4. **Non-repudiation**—it must be possible to prove legally that the message was actually sent and received.

The basic tool to support privacy and integrity is encryption. Data to be transmitted is converted into a different form, or encrypted, such that someone (or some computer) who is not supposed to access the data cannot decrypt it. So, if data is intercepted while en route between Internet nodes, the interceptor cannot use the data because he or she cannot decrypt it. Both encryption and decryption are done with a key and a process (applying the key to the data). Encryption was developed long before the Internet ever existed. Julius Caesar crudely encrypted the messages he sent to his field generals while at war. Until the middle 1970s, the same key was used for both encryption and decryption, so the initial problem was how to transmit the key from the sender to the receiver. This problem was solved in 1976 by Whitfield Diffie and Martin Hellman of Stanford University, who developed public-key encryption, a process in which a public key and a private key are used, respectively, to encrypt and decrypt messages. A communicator—says, Joe—has an inversely related pair of keys, one public and one private. The public key can be distributed to all organizations that might send Joe messages. All of them can use the public key to encrypt messages to Joe, who can decrypt the messages with his matching private key. This arrangement works because the private key need never be transmitted and also because it is virtually impossible to decrypt the private key from its corresponding public key. The technical wording for this situation is that it is “computationally infeasible” to determine the private key from its public key. The most widely used public-key algorithm is named RSA, developed in 1977 by three MIT professors—Ron Rivest, Adi Shamir, and Leonard Adleman—the first letters of whose last names were used to name the algorithm. Most large companies now use RSA for e-commerce. Another, completely different security problem for the Web is the intentional and malicious destruction of data on computers attached to the Internet. The number of different ways this can be done has increased steadily over the life span of the Web. The sheer number of such attacks has also grown rapidly. There is now a continuous stream of new and increasingly devious denial-of-service (DoS) attacks, viruses, and worms being discovered, which have caused billions of dollars of damage, primarily to businesses that use the Web heavily. Of course, huge damage also has been done to home computer systems through Web intrusions. DoS attacks can be created simply by flooding a Web server with requests, overwhelming its ability to operate effectively. Most DoS attacks are conducted with the use of networks of virally infected “zombie” computers, whose owners are unaware of their sinister use. So, DoS and viruses are often related. Viruses are programs that often arrive in a system in attachments to e-mail messages or attached to free downloaded programs. Then they attach to other programs. When executed, they replicate and can overwrite memory and attached memory devices, destroying programs and data alike. Two viruses that were extensively destructive appeared in 2000 and 2001: the ILOVEYOU virus and the Code Red virus, respectively.

Worms damage memory, like viruses, but spread on their own, rather than being attached to other files. Perhaps the most famous worm so far has been the Blaster worm, spawned in 2003. DoS, virus, and worm attacks are created by malicious people referred to as hackers. The incentive for these people apparently is simply the feeling of pride and accomplishment they derive from being able to cause huge amounts of damage by outwitting the designers of Web software systems. Protection against viruses and worms is provided by antivirus software, which must be updated frequently so that it can detect and protect against the continuous stream of new viruses and worms.

**Cloud Computing**

**Cloud computing:** is the delivery of computing services over the Internet. Cloud services allow individuals to use software and hardware that are managed by third parties at remote locations.

**Cloud Computing** refers to **controlling** and **accessing** the applications online. It offers online data storage, infrastructure and application.

**Cloud computing attributes**

**1-Shared resources:**

Cloud computing resources are shared (i.e., multiple users use the same resource) at the network level.

**2- Massive scalability**

Cloud computing provides the ability to scale to tens of thousands of systems, as well as the ability to scale bandwidth and storage space.

**3- Elasticity**

Users can increase and decrease their computing resources as needed and release resources for other uses when they are no longer required.

**Benefits of cloud computing:**

**1- Scalability:**Rather than having to buy, install, and configure new equipment, you can buy additional CPU cycles or storage from a third party. As you require more capacity, the service provider can make scalability much simpler and without cost.

**2- Simplicity:** The cloud solution makes possible to get your application started immediately, and it costs a fraction of what it would cost to implement an on-site solution.

**3-Security:** There are plenty of security risks when using a cloud vendor, but reputable companies strive to keep you safe and secure.

**4- Access to resources:** The greatest advantage of cloud computing is the access it provides to the processing power of multiple remote computers.

**5- Mobility:** Customers can access the services from almost any location in the world because the services are web-based. This can enable employees to access important business tools while they are on the move.

**6- Maintenance and Support:** The supplier will usually offer ongoing support services. The supplier can handle backups, updates and upgrades automatically and remotely without visiting a customer’s site.

**Limitations of cloud computing:**

**1- Your Sensitive Information**: We’ve talked about the concern of storing sensitive information on the cloud, but it can’t be understated. Once data leaves your hands and lands in the lap of a service provider, you’ve lost a layer of control.

**2- Applications Not Ready:** In some cases the applications themselves are not ready to be used on the cloud. They may have little quirks that prevent them from being used to their fullest abilities

**3- Developing Your Own Applications**: Developing your own applications can certainly be a problem if you don’t know how to program, or if you don’t have programmers on staff. In such a case, you’ll have to hire a software company (or developer).

**Security Benefits:**

**1- Centralized Data**

There are some good security traits that come with centralizing your data. Just in practice, you make your system more inherently secure.

**2- Reduced Data Loss**

If a computer can be effectively compromised, the information will be in the hands of the thief. By maintaining data on the cloud, cloud computing can limit the amount of information that could potentially be lost.

**Cloud** **computing** **services**

Cloud computing services goes by the acronym “SPI.”. This acronym (SPI) stands for the three major services provided through the cloud**:**

1. **Software-as-a-service (SaaS)**
2. **Platform-as-a-service (PaaS)**
3. **Infrastructure-as-a-service (IaaS)**

**Software-as-a-Service model(SaaS)**

* SaaS: allows using software applications as a service to end users. Such as Google Docs. Benefits of a SaaS model include the following:

1. SaaS enables the user to outsource the hosting and management of applications to a third party.
2. SaaS enables software vendors to control and limit use (copying and distribution, and facilitates the control of all derivative versions of their software.
3. Applications delivery using the SaaS model typically uses the one-to-many delivery approach.
4. A typical SaaS deployment does not require any hardware and can run over the existing Internet access infrastructure.

**Platform-as-a-Service model(PaaS)**

* PaaS provides the runtime environment for applications, development & deployment tools, etc**.**

**Benefits**

1. Lower administrative overhead
2. Lower total cost of ownership
3. Scalable solutions
4. More current system software

**Infrastructure-as-a-Service (IaaS)**

* **IaaS:** Provides access to fundamental resources such as physical machines, virtual machines, virtual storage, etc. A part from these resources, the IaaS also offers:

1. Virtual machine disk storage
2. Virtual local area network (VLANs)
3. Load balancers
4. IP addresses
5. Software bundles

**benefits others of IaaS :**

1. Full Control of the computing resources through Administrative Access to VMs.
2. Flexible and Efficient renting of Computer Hardware.
3. Portability, Interoperability with Legacy Applications.

**Cloud Deployment Models**

Cloud can have any of the four types of access:

1. Public
2. Private
3. Hybrid
4. Community.

**Public Clouds**

* Public clouds (or external clouds):-The Public Cloud allows systems and services to be easily accessible to the general public. e.g., Google, Amazon, Microsoft offers cloud services via Internet.
* A public cloud is hosted, operated, and managed by a third-party vendor from one or more data centers. The service is offered to multiple customers over a common infrastructure.
* Public cloud may be less secure because of its open source e.g., e-mail.

**Benefits**

1. Cost effective
2. Reliability
3. Flexibility
4. Location independence
5. High scalability

**Disadvantages**

1. Low security
2. Less customizable

**Private clouds**

Private clouds (internal clouds):- used to describe offerings that emulate cloud computing on private networks

There are types of private cloud patterns:

1. Dedicated: Private clouds hosted within a customer-owned data center or at a collocation facility.
2. Community: Private clouds located at the premises of a third party; owned, managed, and operated by a vendor.
3. Managed: Private cloud infrastructure owned by a customer and managed by a vendor

**Benefits**

1. Higher security and privacy
2. More control
3. Cost and energy efficiency

**Disadvantages**

1. Restricted area
2. Inflexible pricing
3. Limited scalability

**Hybrid Clouds**

A hybrid cloud environment consisting of multiple internal and/or external providers is a possible deployment for organizations.

**Benefits**

1. Scalability
2. Flexibility
3. Cost efficiencies
4. Security

**Disadvantages**

1. Networking issues
2. Security compliance
3. Infrastructural dependency