Nuesoft™

The Evolution of Software

Application Service Provider (ASP) Model

Client/Server Software Architecture

Licensed software, enterprise systems, legacy systems

A central database server is connected to personal computers within one location. Functionality, processing and data management are divided between the database and desktops.

- *Complex implementation.* There are hundreds of combinations of hardware and software that might apply, so developers must find the right level of customization.
- *High cost of ownership.* Clients pay for hardware, customization of software to specific hardware packages, installation, and maintenance.
- *High IT involvement.* On-site IT presence required for system upgrades, backups and ongoing technical support.
- *Low scalability.* Not suited for lots of users and there is overhead from supporting the many hardware combinations available over time.
- *Low disaster-readiness.* Data is vulnerable to any on site natural or manmade disaster.
- *Expensive support.* Customers pay for vendor's need to support multiple versions of software.
- *Very low flexibility.* Compatibility between hardware, software and operating systems is required, and may limit installation options.
- *Speed* - varies. Speed is contingent upon the load balance between the server and the computers in the network. As workstations are added, processing times will increase.
- *Can be very secure* if adequate physical operational security measures are taken.

Terminal Services

Thin client computing, Independent Computing Architecture (ICA), pixel pushing

In this hybrid ASP model, a local utility (e.g. Citrix®, Microsoft® Terminal Services) gets visual access to the application.

- *Simpler implementation* than client/server because there is no need for software installation on individual desktops, but on-site IT instead must install and configure software onto the central server.
- *More expensive* than a client-server. Leased, rather than purchased. Initial cost of ownership may be low, but total cost of ownership is high and increases over time.
- *Low involvement of IT.* Eliminates the need for IT to install the application, and centralizes upgrades and maintenance to a few rather than many individual computers.
- *Extremely limited scalability.* Server capacity must be adjusted as users are added.
- *Low disaster-readiness.* Ill-suited for mission critical applications. Although a hosting company may provide remote backup of the data, access to the application is limited if the main servers go down.
- *Relatively expensive support.* This type of technology is costly to maintain, and as the company grows, costs are passed on to clients.
- *Relatively low flexibility.* May limit number of concurrent users, and performance can not be guaranteed.
- *Speed* - varies. Depends on the load on the particular server that is being accessed by the user at a given time.
- *Low security.* This model may violate HIPAA and other privacy regulations, and is vulnerable to hackers if stringent security protocols are not followed by onsite IT professionals.

Web-based

Internet-based

Software as a Service (SaaS), on-demand software; vendor hosts and deploys information from a central application to a large number of users, via the Internet.

Web-based SaaS models rely on the World Wide Web, and thus function using a Web browser.

- *Simple implementation.* Can typically be handled quickly, provided that there is access to an Internet connection.
- *Low initial and ongoing cost of ownership.* Because the technology is based on a distributed model that optimally uses the underlying resources, it costs less to operate the application.
- *Minimal IT involvement.* The vendor handles all upgrades, backups, and ongoing support.
- *Highly scalable.* Can be deployed easily to thousands of users from remote datacenters.
- *High disaster-readiness.* Replicated computing centers at various locations contain complex data redundancy systems and fault tolerance measures.
- *Support is ongoing and constant*, and service level agreements are guaranteed by support.
- *Flexible.* Can be accessed from anywhere there is an Internet broadband connection.
- *Speed* is dependent on the speed of the Internet connection.
- *Generally secure,* as security protocols are handled centrally by a vendor. However, browser-based applications rely on public Web sites, and thus are prone to viruses on the public network, or hackers who are familiar with (and can thus compromise) browser security protocols. Also, applications may rely on security compatibilities between a version of software and browser. If these are not kept current, the application is not secure.

Internet-based SaaS models operate by routing data directly via a virtually private Internet connection.

- *Same simple implementation as Web-based application.*
- *Like Web-based applications, low initial and ongoing cost of ownership.*
- *Similar to Web-based applications—there is minimal IT involvement.*
- *Like Web-based applications, Internet-based applications are highly scalable.*
- *Same high disaster-readiness as Web-based application.*
- *Similar to Web-based applications, support is ongoing and constant.*
- **The most flexible.** Internet-based applications can run on any operating system, and are not limited by versions of browsers or software.
- **The fastest.** Also relies on an Internet connection**.** However, applications that run independent of a browser store a small amount of unique user information on each workstation the first time a user logs in, and so subsequently will run with the feel and response time similar to any other desktop application.
- **The most secure.** Internet-based applications create their own versions of a virtual private network (VPN) over the public network. This private platform between users and their data vastly decreases vulnerabilities to viruses, hackers and other malicious attackers.

	Web	Internet
Flexibility	🚶🚶🚶	🚶🚶🚶🚶
Speed	🚶🚶🚶	🚶🚶🚶🚶
Security	🚶🚶🚶	🚶🚶🚶🚶

1980s → Late 1990s → Today