

Evolution of the ASP Model

Or

Back to the Future

By

Martin J Alfonsi, MDI

April 2, 2001

What is an Application Service Provider (ASP)?

Application Service Providers (ASPs) are entities, which deliver and manage applications/computer services to multiple users from remote data centers usually via the Internet or Private Network.

Why Back?

The current hype surrounding ASPs lend credence to the phrase, "there is nothing new under the sun". Although the term is new, the concept was born in the early days of business computers.

The General Electric Company in the 1960s, allowed individuals and organizations to subscribe to a "time sharing" service. Users would employ high-speed terminals (10 - 30 cps teletype printers) equipped with "acoustic" modems and "dial in" to a GE Mainframe computer somewhere in the United States. The initial service permitted the user to write and execute simple BASIC programs. A library of pre-written routines that performed complex tasks such as sorting, annuity calculations and statistical standard deviation computations soon augmented this. Users could incorporate these library routines into their own programs and could save data for use later. Soon subscribers were performing simple payroll, inventory and general ledger functions.

Complex mainframe computers were large, expensive and beyond the reach of all but the largest companies. Soon many businesses "time-shared" on mainframes managed by "service bureaus" that rented processor time for the relatively mundane tasks of payroll, receivables, payables etc. These early service bureaus were the first ASPs as defined above in all aspects except that the delivery mechanism was not the Internet.

Service Bureaus were able to:

- Concentrate expertise to reduce the cost of application management.
- Provide a location and infrastructure for large complex machines.
- Minimize the cost to the end-user by spreading the enormous cost over a large base of subscribers.
- Insure the safety of important data by performing regular back-ups with secure off-site storage.

It wasn't all rosy however; there were some drawbacks such as:

- Long distance phone calls and/or leased lines were expensive (remember, this was before the Baby Bells and 7 cents a minute)
- The software rental rate was only low if NO changes were made. Customizing was an expensive undertaking.

As technology advanced through the era of mini-computers, micro-computers and personal computers the era of the "time sharing service bureau" passed as companies brought applications first in-house and eventually to the desk top.

Time-sharing did not end because it did not work, but rather the reduced cost of ownership of in-house computing ended its feasibility.

The Desktop Rules

The low cost and high performance of the PC brought power to the average desktop that far exceeded the power and capacity of all but the largest mainframes. Software engineers who for years had struggled to write tight efficient code that stored compact non-redundant data were now free to use all the RAM they needed and fill all the disks they could partition. The claims of the US Justice Department notwithstanding, Microsoft provided a mechanism and frame work that encouraged software development and re-use. Libraries of DLLs grew permitting programmers to spread into new previously unreachable areas. Software became more complex, more user friendly, more graphic, more usable. Mac and Windows allowed the user to buy a computer off the shelf take it out of the box and plug it in. No raised floor, no special wiring, no gazillion ton air conditioner.

However, as users used, they realized they had to/wanted to share. Share information, share data, share peripherals. Computers were connected to each other in twos and threes, in tens and twenties and the Local Area Network (LAN) was born. And along with LANs came "Novell Certified Engineers", highly paid individuals adept in the black art of packets and protocols.

Initially desktops shared data among themselves, but soon it became evident that it would be easier to share the data if it were all in one place. So everybody agreed to use one big PC to 'serve' the data to everyone. *Voila!* the file server was born. Novell was ecstatic and Bill Gates couldn't get Windows NT out fast enough.

Everything got complicated in a hurry. Racks of switches and hubs now joined simple desktop PCs. Words like Ethernet, Cat-5 and 10baseT became *lingua franca*. Office managers struggled with security and software releases. The once liberating DLL Library became "DLL Hell!"

Although the cost of the desktop PC fell dramatically, the total cost of ownership rose just as dramatically. The computing world was ripe for a change.

The Universal Server Teamed with the Universal Client

Once upon a time, a simple man saved the life of the King's only daughter. The King was so grateful he asked the man how he could repay him. "Sire" said the man, "I know you enjoy the game of chess. Therefore I would like you to give me one grain of wheat for the first square of the chessboard. And twice as much for the second square or 2 grains of wheat. And twice as much again for the next square or 4, and twice again for the next square or 8 and so on until all 64 squares are filled". The King thought it was a simple request and replied, "You shall have it".

Little did the King realize that in order to fulfill the man's demand it would take all the wheat his kingdom could produce for centuries. This schoolbook story vividly demonstrates the power of geometric progression. But it pales by comparison with the growth of the Internet.

The Internet Software Consortium (<http://www.isc.org/>) listed 213 web host sites in August of 1981,

- 1,000 in '84,
- 5,000 by '86,
- 100,000 by '89,
- 300,000 by '90,
- 1 Million by '92,
- 2 Million by '93,
- 12 Million by '96.

By January 2000 the number of host sites had grown to 72,398,092 and is nearly doubling every twelve months.

"The Internet provides a publicly accessible infrastructure that connects users to off-site application servers." Now with a low cost browser there is a standard way to access applications and information from a variety of host sites. Using this low cost technology, Internet Service Providers (ISPs) and software vendors are rewriting the rules of information management.

Traditional ISPs manage servers that host Web pages and route messages through e-mail servers. An ASP is simply an entity that extends the service to software programs such as payroll, customer service, and accounts receivable. ASP customers can interact remotely with a managed application through a simple low cost web browser anywhere at anytime.

As was the case in the "time sharing" service bureaus, ASPs **rent** software that a company does not desire to buy for themselves.

ASPs are able to:

- Concentrate expertise to reduce the cost of application management.
- Provide a location and infrastructure for hardware server "farms"
- Minimize the cost to the end-user by only requiring low cost client software on their desk top.
- Insure the safety of important data by performing regular back-ups with secure off-site storage.
- Ensure customers are using the latest versions of relevant applications.

ASP customers are able to:

- Add customers or scale upwards without expending capital for costly hardware and implementation plans.
- Implement new applications (ERP, CRM) faster. This translates into a quicker time to market.
- Integrate remote offices quicker. Enterprise applications that were previously not cost effective can now be used.
- React to mergers and acquisitions more efficiently.
- Improve operational freedom by concentrating on their core business functions.
- Streamline budgeting and cash flow management by avoiding costly peaks. Financial flexibility.
- Reduce risk, since without a need to invest heavily in IT and hardware, new techniques can be tested and either implemented or discarded easily.
- Afford to use major applications that would otherwise be too costly.
- Eliminate the need for IT infrastructure to deal with operating systems, product upgrades and complicated license fee schemes.

ASPs charge for their services based on a monthly fee for a fixed period of time, typically between one and three years. There are three basic pricing models in use:

- 1) A per user/month charge, which can range from \$10-\$100 for low-end applications to more than \$5K for complex front/back office applications;
- 2) A fixed monthly fee based upon the amount of computing, networking and labor resources required to service and support the application;
- 3) Transaction-based pricing, where users pay for particular business transactions associated with the application.

All three models provide discounts based on the volume of business and the length of the contract term.

There are problems however, that must be addressed. Among them are:

- Internet congestion that may slow response time, especially on publicly accessible networks.
- Inability to have extensive customization. (The ASP model only makes sense when all subscribers are using the same version of a product.)

Clearly the ASP model to be effective it must and is becoming as ubiquitous and as sophisticated as the public telecommunications system. ASP users must find better reliability and lower costs than comparable in-house systems. This will happen since the ASP is concentrating on the delivery and management of the application provided and not on the business of employing the application.

Although the rebirth of this model is in its infancy, it is becoming clear that in order to be successful and effective, ASP must be in a position to offer:

- Web-site hosting and e-mail storage and routing services
- ERP and other complementary business application hosting
- Configurable e-commerce storefronts for selling on the Web
- Access to procurement trading hubs for buying on the Web
- Support for EDI (electronic data interchange) or other electronic transaction transmission
- Secure Internet access to application servers (e.g., via a virtual private network, or VPN)
- Highly configurable application-level security to limit functional access
- Online training courses, manuals and frequently asked questions (FAQ) documents
- Online application support via real-time Internet chat, e-mail and self-service help desks
- Use of backup servers to provide application uptime 24 hours per day, 7 days per week, 365 days per year
- Automated offline data backup scheduling for disaster recovery
- Automatic load balancing to ensure optimum application accessibility under user load
- Simple "add a new user" Web sign-up
- User-based monthly application rental fee and low initial system setup fee (if any)
- Regular statistics showing user activity by application for usage analysis
- Delivery of application reports, documents and alerts to users via e-mail
- Electronic workflow via e-mail or a Web browser
- Service level agreements that ensure mutually acceptable service standards are met
- Capability to automatically upload data to and download data from applications

Clearly the ASP should not be viewed as the solution for everyone. . . It still makes the best sense for users who do not have the time or resource internally. Initially a tool of the service provider community, a growing number of companies (and investments in companies) are moving towards the ASP model. Until there is another major technology shift, we can expect the Fisher-Pry rule (*new technology develops slowly until saturation reaches 20%, then exponential growth is expected*) to impact the future of the ASP.