APPLICATION SERVICE PROVIDERS: A “NEW” CIO OPTION

Leilani Allen

INSIDE
Definition of ASP; Reasons for Choosing an ASP; Case Example

A HISTORY LESSON
For those who have been in the technology field for some time, the phrase “everything old is new again” is particularly meaningful nowadays. One of the current hot concepts for managing information resources is that of the ASP (application service provider). This notion hearkens back, literally, to the early 1970s, when it was known as “time-sharing.” In those days, computers were expensive and the expertise required to program and maintain them was scarce. Also, companies were not as dependent on computers as they are today. They might only need the computer to run month-end payrolls or inventory reports. So, rather than purchase an entire configuration, they purchased part of one. Much like the borrower who does not need a vacation home at the beach all year-round, but rather for two weeks in July, and decides on a time-share, so too could companies purchase just the computer power they needed when they needed it.

The time-share concept eventually diminished as computers dropped in price, and, especially, as companies went from batch to online processing. Once most of the users were accessing the computers during most of the day, it did become cost-effective to have one’s own data center with one’s own computer staff. That was the heyday of mainframe computing, as companies built proprietary applications designed for their specific needs.

PAYOFF IDEA
Application service providers (ASPs) offer advantages to companies that have too many initiatives and too little talent. They provide an efficient way to utilize hardware, software, network, and human resources. However, the more innovative a firm wants to be, the less likely it will be comfortable with an ASP as a long-term solution.
Along came client/server, which helped decentralize computing and, in many cases, turned over the development of software and the management of computer hardware to the business units. Instead of a $20 million mainframe, a department could now buy a minicomputer or server, for prices that easily fit into its budget. Everyone was gung-ho to create the latest and greatest client/server applications.

So what about the mainframe applications? Over time, a majority of industries had standardized on a couple of operating system/subsystem choices, with IBM’s MVS/ESA and CICS being the leading combination. Once this was accomplished, the care and feeding of the operating environment became largely generic. This meant that knowledge of the particular business was presumably no longer essential.

Thus began the era of outsourcing — when companies large and small outsourced their data center and their core applications to companies such as IBM, EDS, CSC, and a host of system integrators. Outsourcing was particularly welcome in the 1980s, when organizations sought to become “lean and mean,” and eliminate all “unnecessary” headcount. How much easier it was to turn over the whole responsibility to someone else.

Outsourcing took at least three forms. The first model was to outsource all of information technology (IT) — network, hardware, and software development — keeping only a skeleton staff to manage the contract with the vendor and do strategic technology analysis. The most common approach was to outsource the first two components but keep software development in-house, and thus more responsive to the particular business. A third approach was to turn to outsourcers (service bureaus) for specific applications such as accounting or payroll.

As usual, no technology solution is completely satisfactory. Companies soon found that the outsourcers often underbid the contract for the first one or two years of operation, but that, over time, the cost exceeded the original in-house installation. Also, many employees who really knew the applications were lost in the shuffle, making maintenance and enhancement a real challenge. Indeed, one of the reasons that the cost for year 2000 (Y2K) remediation was so large was that so many people who had built the applications in the first place were no longer available.

Unfortunately, one of the downsides of outsourcing for the technology industry was that innovation in mainframe technology largely disappeared. The last thing an outsourcer wants is new and different technology to migrate their clients to.

Thus, outsourcing gave an indirect boost to client/server technology. Firms found that, with their core production (legacy) applications safely in the hands of outsourcers, the remaining IT staff was free to focus on strategic systems that utilized the newest technology. A lot of innovation occurred at the level of departmental applications. True, with innovation came a lot of failures that gave client/server a bad reputation for a while.
As time went by, however, the client/server architectures became more stable and reliable, and IT became more skilled and knowledgeable in this technology as well.

So today, there is a whole new layer of mission-critical production systems that are now built on client/server technology. They are often very complex, involving multiple servers and network connections. They are usually built upon databases that require special expertise to design and maintain. And, as before, there has been a good deal of standardization in the languages and tools used to create these systems.

This is another opportunity to outsource. Oracle Corporation President Larry Ellison has declared his firm to be the world’s largest and fastest-growing ASP. He says, “The software industry is in the process of a huge change, a tectonic change. Software is on its way to becoming a service.” And software companies are becoming service providers.

**DEFINITION OF AN ASP**

An ASP can be defined as a firm that “implements and provides ongoing support of application software for customers on one or more computer platforms and networks.” The term ASP was supposedly coined by International Data Corp. of Framingham, Massachusetts, in 1998 in a study on the future of outsourcing. By May 1999, 25 companies had joined a newly formed ASP Industry Consortium. By November, 178 companies had joined. At the Consortium’s first conference in Denver, Colorado, almost all of the technology industry heavyweights spoke in favor of the concept.

For example, Compaq and Cable and Wireless (a U.K.-based telecommunications firm) announced a global network of data centers that would host applications, and provide help desk services, systems integration, and market strategy. On the software side, Microsoft unveiled Office Online to deliver its popular suite of productivity products to users across an Internet connection. Corel already rents WordPerfect for $9.95 per user per month, while Sun Microsystems began giving away its StarOffice suite last year. Indeed, Gene Banman, a Sun executive, says, “Hosted application services will replace boxed software sales altogether in the next decade.”

Hardware vendors are equally enthusiastic. IBM is repositioning its venerable AS/400 minicomputer as an ASP platform. Hewlett-Packard and KPMG are working to build CyberCenters to host data warehouses. Novell, Inc., recently acquired Just-On, Inc., a Web-based file management business. PC maker Acer, Inc., in Taiwan is partnering with software giant Computer Associates in Islandia, New York, to offer ASP services. Cisco Systems and Sun Microsystems are developing standards and practices guidelines for ASPs, including a certification program.

The latter point is particularly important. Just as there were no strict rules about the services that an outsourcer once offered, there is no strict
definition of an ASP. Many firms do not yet have documented contracts in place, and have not really worked through their economic model. Services, performance, and availability of applications are also fairly restrictive.

There are at least three models of ASPs, in descending order of popularity:

1. **Internet Service Providers (ISPs)** provide access to individuals and firms for Internet services. ISPs vary widely in terms of services provided, but most are still confined to hosting Web sites and providing e-mail capabilities. Market researcher Input in Mountain View, California, expects Internet management outsourcing to grow 76 percent annually between 1998 and 2003. A recent *Computerworld* (October 25, 1999) survey of IT managers found that 22 percent of firms were already using an ISP for Web hosting, although 47 percent said they had no such plans.

2. An emerging group of companies are what this author refers to as **Bundled Service Providers (BSPs)**, essentially service bureaus hosting a handful of interrelated applications. The most popular are enterprise resource planning (ERP) systems from such firms as SAP, PeopleSoft, or Baan; or customer relationship management (CRM) systems that bind together telephone, computing, and Internet technologies to support telemarketing sales and service centers.

3. Finally, there is the **Proprietary Service Provider (PSP)**, usually a software firm that will make its software available on a shared basis. Instead of paying for the software outright and running it on their own computers, customers contract with the PSP to connect to the vendor's architecture. PSPs vary widely on the ancillary services available to customers.

**WHY ASP?**

There are already a number of ASP success stories from firms as varied as Monsanto, Volvo, Fleetwood Retail, Robert Mondavi Winery, and Barnes & Noble. Cynthia Morgan, writing in *Computerworld* (October 25, 1999), finds that ASPs make sense for larger firms when:

1. the application requires expertise lacking in the existing IT staff
2. IT has more pressing projects
3. a neutral party is needed to merge and centralize services
4. an application needs extremely rapid deployment
5. users will be widely scattered, often with only a handful at a particular site
6. users are outside the firewall

Almost all experts agree that the ASP solution is primarily for smaller companies that want to use a world-class application but cannot afford
the up-front costs of acquisition, or the ongoing personnel burden of maintenance. Because they are most susceptible to swings in market conditions, smaller companies like the fact that ASPs usually charge on a variable price basis. If volume diminishes, the monthly bill declines, although the actual cost per transaction (or loan) may increase because volume discounts no longer apply.

Not surprisingly, the cost-effectiveness of ASPs is similar to that of any leasing arrangement. The longer one expects to keep a given vehicle, the less it makes sense to lease, because the cost of ownership can be amortized over a longer period of time. On the other hand, in the world of technology, long-term relationships are not the norm, and a firm can expect to change its technology every three to five years.

Continuing with the car analogy, leasing a car requires the dealer to be responsible for maintenance and repairs, recalls, and model obsolescence. Similarly, it is the ASP that must worry about system maintenance and performance, capacity, and technology obsolescence. The ASP must contend with vendors, dealing with bugs and sweating through the installation of new releases and issues of system compatibility.

Most importantly, the ASP must worry about recruiting and retaining knowledgeable staff. At the moment, there are an estimated 300,000 technology jobs going begging. Even small firms are finding that the cost of IT professionals is beyond their means, with a typical programmer/analyst earning as much as senior managers in the business units, and specialists such as database administrators or Web masters earning the same as executive vice presidents. It is almost impossible for many firms today to find these people, provide sufficient technical challenges and financial incentives to keep them, and have appropriate management to make sure they are doing the right things. Thus, ASPs are an attractive alternative.

Another factor fueling ASPs is the fact that implementation times can be a lot quicker. The organization need not build an infrastructure to handle the application; the infrastructure is already in place. A firm’s representatives can visit the service bureau, see the servers and network configurations already in place, presumably happily serving other customers. More and more, ASPs are also providing an electronic commerce (E-commerce) infrastructure for companies that want a fast path to Internet viability.

The argument in favor of ASPs is neatly summed up by Craig Kinyon, CFO for Reid Hospital and Health Care Services, Inc., in Richmond, Indiana: “The ASP model takes care of what I call the ‘Tylenol factors’ of applications — the maintenance, support, upgrades, and hardware. And we don’t have to find and pay for staff to handle the applications.” (InformationWeek, February 21, 2000)
A CASE EXAMPLE

A small mortgage company (less than 300 employees) has offered government (FHA/VA) and conventional (Fannie Mae/Freddie Mac) loans primarily to first-time homebuyers in the northeastern United States. Its loan origination and servicing technology was written by a third-party provider to run on a DEC platform. Two or three staff members were hired to work with the third party to run the hardware and maintain the software.

Mortgage lending is an industry where there are a great number of regulatory changes — not exciting work, but necessary. The IT staffers who were brought on board quickly became bored and left. Maintenance of the hardware was outsourced to a contractor; maintenance of the software largely ceased because it was too difficult to find the right technical skill set. The third-party provider (essentially a one-man shop) moved on.

To compound matters, this is a period when mortgage lenders are fundamentally reexamining their processes and approach to the marketplace, wishing to include such facilities as workflow and Internet lending. To stay competitive, the firm must update its technology. Needless to say, the existing platform will not accommodate such profound enhancements.

The firm evaluated third-party mortgage lending packages and found three that had the functionality it desired. All the packages used relational database technology and ran on NT servers. However, there would still be some customization required. So, once again, the firm was confronted with the task: Who would customize and maintain the software on a go-forward basis? Where would the firm get the NT expertise to run the hardware?

One of the software firms raised the concept of an ASP. It offered to customize the software and maintain it, as well as operate the hardware (the database server from its own data center, and the application server on site). In addition, the vendor could provide database, help desk, security, and disaster recovery services. The cost? A setup charge, as well as a fee per user per month.

To the mortgage lender, it seemed an attractive alternative. The firm would overcome its personnel problem by having the vendor’s staff perform the work. Recruiting, training, management, and retention would all be someone else’s problem. The vendor would be responsible for regulatory and other software modifications, as well as keeping abreast of technology changes.

The executives liked the idea that they would not have to pay the full cost of the hardware and software up front (approximately $1 million), but rather could “lease” the technology on a pay-as-you-go basis. This is especially important in a highly cyclical industry such as mortgage, where interest rate fluctuations bring feast or famine volumes. Assuming that the number of users generally reflects the volume of work, the firm
could translate its technology cost from fixed (if it owned the technology) to variable.

Not everyone was pleased with the proposal. Some of the managers were concerned about not having on-site support. They also did not like the fact that the firm would not own the software, but merely lease it. The hardware contractor began a campaign of denigrating the supposed reliability and performance of NT.

Moreover, the vendor’s sales representative was only lukewarm to the concept. Part of this was lack of familiarity; part of it was the impact on their wallet. After all, salespeople are traditionally paid commission based on the value of the signed contract. In an ASP mode, the total amount of the sale is unknown; commission would be calculated on an amount to be paid out over time based on usage. Not as attractive a proposition. This is a factor that software-companies-cum-ASPs will have to consider.

Thus, a firm that would appear to be an ideal candidate for an ASP still has hesitations about whether the model will work. And the vendor still has some homework to do on selling its own people.

THE FUTURE

Will every technology company become an ASP, or call itself one? Probably, for a while. However, a number of software firms will undoubtedly find that running reliable “data (cyber) centers” is more difficult than imagined. For one thing, the ASP concept calls on a different set of expertise than what they have used to develop whiz-bang applications. Many of today’s current software developers have little understanding of, or appreciation for, such issues as performance analysis, tuning, and capacity planning. In the past, if the application grew too big or obstreperous, the solution was simple — buy a bigger box. To maintain profitability, ASPs will not be able to simply resort to this practice. They will need to have a rigorous program of configuration management that is constantly evaluating and improving the reliability and performance of their hardware and networks.

And, as recent experience has shown, the software firms-come-ASPs will have to take the same rigorous approach to security management that was the norm for the big outsourcers, with multiple layers of security and disaster recovery. After all, an individual firm’s computer center was not much of a target for hackers; however, an ASP handling business for dozens or hundreds of firms is far more inviting. Indeed, the Computerworld (October 25, 1999) poll referred to above found that 96 percent of respondents had concerns about security. On the other hand, an ASP is more likely to have access to security experts than a typical small company.

Also, as anyone who has ever endured the torture of calling a software provider’s customer support line knows, these firms have a dismal
record in post-market support. Staffing a competent help desk is no easy task, but will be mandatory for a successful offering as an ASP. The firms are hosting mission-critical applications, and users will need quick and accurate answers. Yet finding the source of a problem will prove very difficult for those ASPs who try to integrate software from several different vendors.

Another key problem for an ASP firm is finding the appropriate price model. For example, CenterBeam, Inc., in Santa Clara, California, is offering a Lucent Technologies-based network, a PC running Windows 2000 and Microsoft Office applications, a wireless LAN service, a public Web site, a company intranet, high-speed DSL Internet access, daily data backup, and 24×7 technical support for less than $200 per user. BSP firms will have higher prices than this because of the database and storage costs for high-end applications.

Still, Forrester Research of Cambridge, Massachusetts, expects the ASP market to jump from $150 million in 2000 to $6 billion in 2001. Dataquest in Redwood City, California, is far more optimistic, predicting $22 billion in revenues by 2003.

EVALUATING ASPs

Not every ASP will be successful. How does one know if one is dealing with the right firm? Generally speaking, it is better to deal with an ASP that offers a focused group of applications. This will be a learning experience; and the greater the number of variables, the greater the chances of failure. It is important to find an ASP that has developed the offering as part of a natural evolution of its product or industry, rather than one that is just a case of “me too.” The range of issues to consider is lengthy, but a preliminary list includes:

• **Expertise.** Just what are the provider’s qualifications? Has it built infrastructures before, or is it simply a software developer? Has it run a 24×7 operation? What is its motivation in doing this? How does it expect to make money? Will the provider be in it over the long haul? Obviously, the greater the expertise, the better. Also, the financial expectations should be realistic and should include an honest assessment of the costs. And it is essential that the provider has an understanding of system performance and capacity planning.

• **Service.** When is the system available? What response time is guaranteed? What happens if service degrades? What happens if there is a system failure? The more competent the ASP, the more guarantees of service it will offer, and the more likely it will be to offer compensation for its errors.

• **Help desk.** How are people trained on the help desk? What questions do they handle — technical only or application specific? What is the allowable wait time for a response? Is the response received over the
phone or via e-mail? What hours does the help desk work? What happens after hours? The help desk will be the only real contact most of a client’s users have with the ASP, so support must be of the highest possible caliber.

- **Security.** How is this handled? Will your auditor be comfortable? What happens in the event of a security breech, and who is accountable for damages?

- **Special services.** Does the ASP provide database administration, disaster recovery planning/testing, project management? Is some of this built into the standard contract, or is each service separate? The former method is preferred. Otherwise, one gets the “nickel-and-dime” effect.

- **Licensing.** Who pays, and on what basis? Can software be made “rent to own”? If the intention is to utilize the ASP for less than three years, then negotiating to own the licenses at the end of the contract is important.

- **Scalability.** Can the infrastructure grow to handle your needs? Who decides when upgrades are necessary, and how is payment handled? This is especially important if a number of clients are sharing a configuration.

- **Software maintenance/enhancement (especially for BSPs).** How are new releases handled? What if one does not want to move to the new release? How does one request enhancements, and who gets a vote on setting priorities? How are enhancements tested? It is important that the ASP have a rigorous system life-cycle methodology, and a careful method of deciding which enhancements will be part of the standard release, and which are truly custom. Otherwise, the costs can be prohibitive.

- **Pricing.** Is the charge per user or per transaction, and how is the transaction defined? For example, for the mortgage lender, is the charge per user, per loan application, or per closed loan? The cost models of each industry will differ. And the ASP should provide guidance in helping the client get the most bang for the buck.

- **Implementation.** What is the process for getting up and running? What does the provider do? What does the client do? Is there a sample contract? Is there a significant choice of offerings, and the ability to negotiate a customized deal? Flexibility is good, but beware the ASP that claims to provide all things to all people. Part of the deal in sharing a resource is learning to live with limitations; both the client and the ASP have to understand this. And that will be the most difficult lesson of all.

- **Administration.** How is billing done? One may wish to pay on a quarterly or monthly basis. Can the provider accommodate this? One should ask to see a variety of performance reports each month: availability, response time, help desk call resolution times, etc. The ASP
should have a mechanism in place to routinely survey its customers, and it should be willing to share the results of internal audits. There should be a user group or advisory council in place where customers can routinely influence the ASP’s direction. The more willing the ASP is to shine a light on its own operation, the more confident one can feel in its capabilities.

- **Termination.** There is no point in maintaining a relationship if one partner is unhappy. As much as the ASP will want to lock one in for the long term, there should be a relatively painless and graceful way of terminating the contract if expectations are not met. Look for provisions that provide reasonable notice (no more than 90 days) and reasonable penalties (no more than the equivalent of six months’ lost revenue, but try for something even better).

**CONCLUSION**

ASPs offer clear advantages to firms large and small that are contending with the problem of too many initiatives and too little talent. ASPs offer an efficient way to utilize hardware, software, network, and human resources. However, a shared platform often results in catering to the lowest common denominator. Therefore, the more different or innovative a firm wants to be, the less likely it will be comfortable with the ASP as a long-term solution.

The best advice? Perform due diligence with particular care, and find a vendor that has good solid business reasons for being an ASP, as well as a plan that makes it a long-term player. If one decides to take the plunge, take advantage of being an early customer by ensuring that one has a voice in how the ASP is run, and how it develops its products, prices, and services over time. And monitor, monitor, monitor the ASP’s performance — not only in the reports it provides, but by surveying one’s users. Remember that — outsourced or insourced — to your business peers, technology is still your responsibility.

Leilani Allen, Ph.D., is a partner with Summer Point Consulting in Mundelein, Illinois. She can be reached at leilani@flash.net.