Software as a Service: Strategic Backgrounder

February 2001

Software & Information Industry Association
1730 M Street NW Suite 700
Washington, DC 20036
+1.202.452.1600
www.siia.net

Acknowledgements

This document grew out of discussions between SIIA staff and a growing contingent of association members who are either evaluating or entering the Software as a Service (SaaS) market. Contributing to this effort from SIIA include Fred Hoch, Director, eBusiness Division, Michael Kerr, Manager and Anne Griffith, Director of Research. We would like to thank all the members of SIIA's Software as a Service Initiative for providing input. SIIA would like to acknowledge in particular the significant contributions of David Greschler and Jeff Dronzek of Softricity, Mark Reichert of Wizmo and Rick Weir of Online Technologies Corp.
.

Table of Contents

Introduction

Software as a Service (SaaS), commonly referred to as the Application Service Provider (ASP) model, is heralded by many as the new wave in application software distribution. Following the maxim that "the Internet changes everything," many believe that traditional packaged desktop and enterprise applications will soon be swept away by the tide of Web-based, outsourced products and services that remove the responsibility for installation, maintenance and upgrades from over-burdened MIS staff. Some analysts and industry members believe that packaged software, as a separate entity will cease to exist. While such drastic predictions have not yet happened, due to technical and business issues, the spirit of this change – the delivery, management and payment of software as a service rather than a product – is affecting all participants in the software industry.

SIIA has developed this backgrounder to analyze the current state of the SaaS market and its near term prospects, and to provide insight for our members who may be profoundly impacted by changes implied in the SaaS model. As an association representing traditional software publishers, ASPs, service and infrastructure providers, SIIA is uniquely positioned to examine the competing forces promoting and resisting the model.

Defining Software as a Service

SIIA believes that the software as a service model is capable of causing a sea change in the software industry. However, Software as a Service still must overcome several significant obstacles to widespread adoption. The first among such obstacles may be the lack of clarity in the definition of software services themselves. The market is hindered by disagreement over the intrinsic characteristics of services and even the terminology used to describe application services. The definitions are constantly shifting, buffeted by the creation of new business models and technologies that companies employ to deliver their vision of software as a service. The marketplace is inundated with acronyms each representing a slightly different approach - application service provider (ASP), application infrastructure providers (AIPs) Internet business service (IBS), business service provider (BSP), solutions service provider (SSP) and more. Therefore, to avoid confusion SIIA refers to the model generally as software as a service.

In the software as a service model, the application, or service, is deployed from a centralized data center across a network - Internet, Intranet, LAN, or VPN - providing access and use on a recurring fee basis. Users "rent," "subscribe to," "are assigned", or "are granted access to" the applications from a central provider. Business models vary according to the level to which the software is streamlined, to lower price and increase efficiency, or value-added through customization to further improve digitized business processes.

The core value of software as a service is providing access to, and management of, a commercially available application. The potential benefits of the model are significant for both the vendor and the customer. This service is different from business process outsourcing (BPO), for instance, where the outsourcing contract encompasses management of entire business processes such as HR or finance. It is also different from hosting services, where the focus of the service is management of the network and servers, but virtually no applications management.

Originally coined by IDC, the term "Application Service Provider (ASP)" referred to companies that met a strict set of defining characteristics. These included the following[1]:

- *Application centric.* The core value of the ASP service is providing access to and management of an application that is commercially available.

[1] *Source: "ASPs Are for Real ... But What's Right for You?" An IDC White Paper (www.idc.com). Reprinted with permission.*

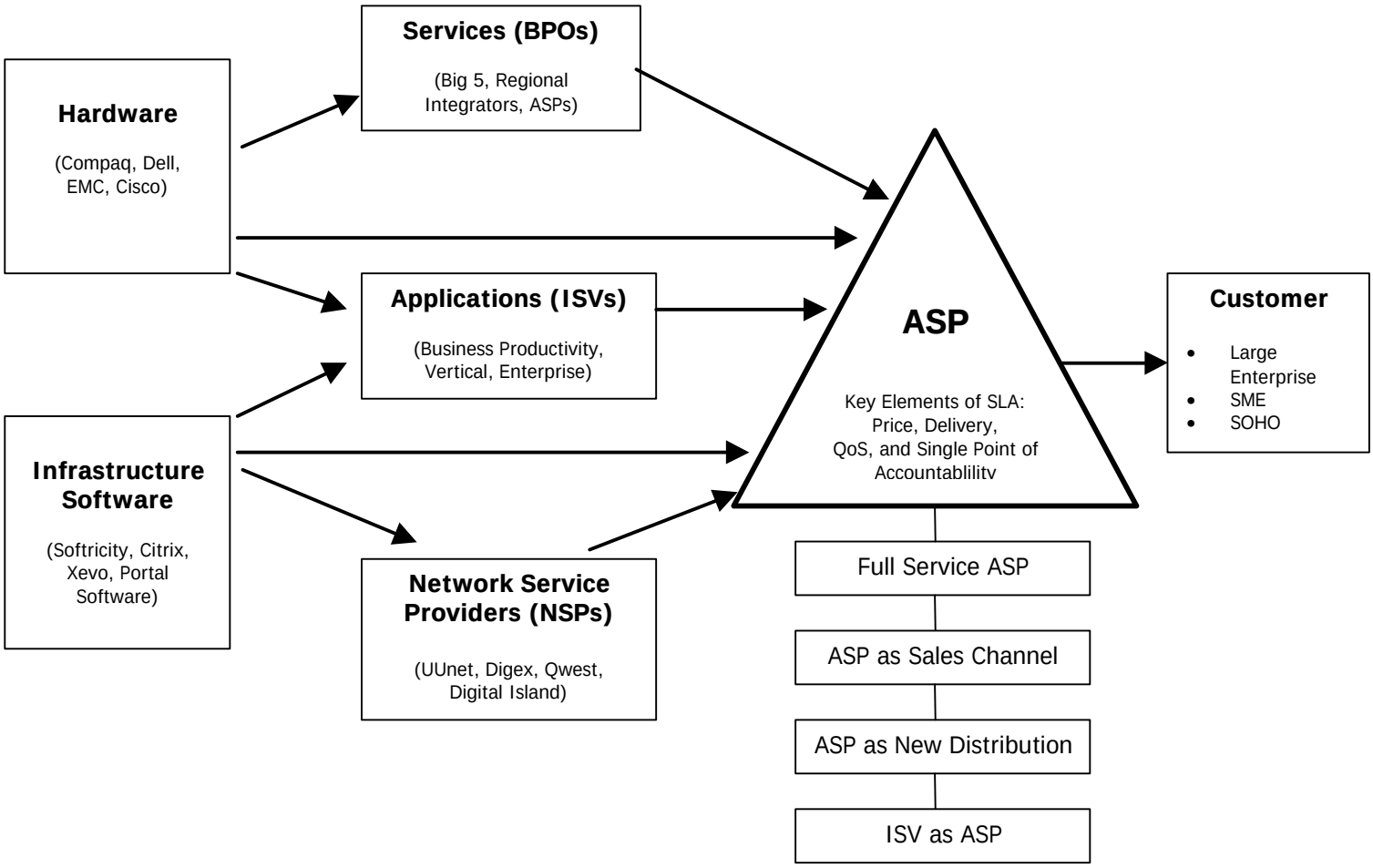
- *An ASP "sells" the application access.* Part of the value of the ASP services is that customers gain access to a new application environment without making up-front investments in the application license, servers, people, and other resources. The ASP is able to add this value to these services either by owning the software or having a contractual agreement with the software vendor to license access to the software as part of the ASP's offering.
- *Centrally Managed:* The application service is managed from a central location rather than at each customer's site.
- *One-to-many service*. The ASP service is designed to be a one-to-many offering.
- *Delivers on the contract*. There are many partners working together to provide and ASP Solution. The ASP is the firm that is responsible, in the customer's eyes, for delivering on the customer contract; that is, seeing that the application service is provided as promised.

However, since the term was originally coined, a number of companies who do not meet these criteria have begun to refer to themselves as ASPs. Companies that provide any outsourced application services, or component of the process, call themselves an ASP. The result is a great deal of confusion in the marketplace.

Companies calling themselves ASPs include those identified by IDC (e.g. USinternetworking) as well as traditional ISVs who have begun to offer their products in a new manner (e.g. Intuit). New companies, Internet Business Service Providers (e.g. Agillion.com, NetLedger.com), are developing software that is native to the Internet from the onset. Moreover, value-added resellers and integrators are beginning to host services and manage the software that they had previously installed on corporate networks. Furthermore, Aggregators (e.g. Jamcracker) that serve as an integration point and front end for customers wishing to rent applications from multiple service providers have arisen. In our view, Software as a Service encompasses the companies and offerings that now make up the much broader definition of this business model.

The diagram of the ASP Value Chain on the next page details a simplified view of the evolving component parts of the software as a service marketplace. Hardware providers are supplying the servers and networking components necessary to make the model feasible. A new breed of "Software Infrastructure" providers are creating a platform for delivery of software applications as well as facilitating the conversion of traditional boxed application of the ISV to a networked service offering. Network Service Providers are furnishing the pipes and consultants are integrating the new services and data with legacy systems. All of them are lending support in creating ASPs in a variety of forms, each dependent on the needs of the business model and how it fits with the company's historical revenue models. For some ISVs, an ASP model is just a new sales or distribution channel not a wholesale revamp of the company's strategy.

THE ASP VALUE CHAIN



Moreover as the market evolves, the offerings that define companies are blurring. In the emerging Software as a Service market, "pure play" ASPs are beginning to get involved in application customization because their customers require it; and taking application infrastructure portions (network, delivery, etc.) out from the customer interface. For instance, Corio -- which once offered both infrastructure and customer application interface development -- is now focused exclusively on the customer interface, leaving the infrastructure to the AIPs.

Secondly, a host of companies other than "pure play" ASPs are offering software as a service. First and foremost are the ISVs, who, not wanting to lose their direct relationship with the customer, are transforming themselves into "ASPs." Other companies rolling out software as a service include system integrators, who are changing their core services from integration (the customer owns the application and delivery infrastructure) to service (customer rents the application and the integrator manages the infrastructure.) Finally, there is a pronounced trend within large enterprises to transform the IT department into an "internal ASP," effectively acting as a third-party provider of services.

While pure play ASPs were the first to offer software as a service, the bigger industry trend is for all elements of the software chain – plus the network providers who are new additions to the delivery chain – to begin offering software as a service.

Independent Software Vendor Approaches to Software as a Service

The Business Case

Among the strongest proponents of the Software as a Service concept have been many of the largest, most influential enterprise software publishers. The concept is a significant departure from the 'development and release' business model familiar to this group and their corporate and individual customers. But it is a departure that comes with attendant costs and benefits. Two frequently cited costs in migrating to an online service include committing to more frequent product releases and upgrades and sacrificing the recognition of large, up-front license payments for a recurring rental revenue which may be smaller at the outset.

Why, then, are so many software developers considering providing software as a service? There are three primary benefits to the vendor:

1. Reducing the substantial costs of code delivery to the customer. Hosting the application means that service providers can offer faster application modifications to customers with potentially less debugging work than before.

2. Recurring-revenue charge models such as those enjoyed by ISPs are extremely appealing to developers. Even installers of traditional ERP applications, who are accustomed to large up-front application customization charges but smaller annual maintenance fees, may be attracted to the SaaS model as a way to smooth revenue bumps.

3. Web-based services model greatly expands the potential customer base for the ISV, offering access to the entire Internet business market. Since growth for high-end ERP applications among Fortune 1000 companies has stagnated as the Tier 1 market has become saturated, these new markets are vital to continued growth.

There are many other good reasons for moving to a software service model. Vendors may want to:

- Move a company with a traditional business model to a more Internet focused approach.
- Offer better online services and information to customers.
- Limit the costs of configuring software for customers, suppliers and internal users.
- Attract new investors and partners.

ISV Market Entry Strategies

Software publishers are approaching the market in a variety of ways. Some developers adapt current products or incorporate SaaS elements into existing product lines and host/manage the services themselves. Some re-tool their entire product lines - albeit in stages - to create new products built for the ASP model, while some roll out completely new offerings. Hundreds of other companies have been established in the past few years specifically to provide software in this fashion.

Most ISVs have announced strategies or offerings that involve a direct SaaS solution or partnerships with pure-play ASPs, or both. The following table offers an abbreviated look at how several leading publishers are approaching the service model.

Direct SaaS Solution[2]	Partnership with ASPs, AIPs and/or Integrators	Combination	SaaS Enabler	SaaS Management & Support
Oracle	Great Plains	SAP	Citrix, Softricity	Xevo, CyberSource
Intuit	Siebel	J.D. Edwards	Tarentella, GraphOn	Portal Software
	Lawson	PeopleSoft	New Moon	BMC Software

[2] *Some direct ASP players partner with infrastructure providers*

As a leader in the direct solution market, Oracle has created its Oracle Business Online subsidiary to attack the market. Oracle Business Online operates like a utility in that it gives companies - especially smaller companies and start-ups - rich application functionality at often dramatically lower costs than they would incur to license, install, and maintain the software in-house. Through Oracle's preferred infrastructure partners, Oracle Business Online manages customers' applications from its network of highly secure data centers worldwide.

Intuit provides another example of a direct to market SaaS solution. The company has refocused its sales of TurboTax from boxed software to services online. Intuit created web-enabled, web-accessed versions of its previously packaged software products. It has been a profitable move. Services such as TurboTax.com and Quicken.com have seen software service revenue increase by 106% year-over-year and now account for 35% of Intuit's total revenue. Other traditional software companies are following suit.

Combining both direct strategies with partnering arrangements, SAP has created an operating subsidiary, mySAP.com Application Hosting, and has certified pure-play ASPs to deliver its software as a service (e.g. EDS, eOnline, Qwest Cyber.Solutions). Like SAP, J.D. Edwards has created its own ASP subsidiary and named several certified pure-play ASP partners. However, the dual models hold the potential to create significant channel conflict as the ISV may compete with its partners for the same customer. Other smaller software firms are choosing to continue their normal sales and marketing strategies and if the customer wants to pursue an ASP model, they will partner with an enabler.

Most companies selling software as a service are members of a supply chain (see chart on page 6) that comprises several vendors and network service providers. Like outsourcing, this system moves the responsibility of providing the real estate, human expertise, and automation to run a service— in this case, the provision of applications— from internal IT operations to a third-party. The ASP will upgrade server-processing power as needed, install patches and upgrade to new versions of the software. They will also procure network connections to the hosted applications on behalf of the customer and take responsibility for the specified service-level (SLA) of network on-service time.

SaaS Enablers and Supporting Technologies

Some software vendors have taken a different tack to approach the SaaS opportunity, employing technologies that enable them to turn their products into services. Vendors like Citrix, Softricity, GraphOn and New Moon are offering application delivery products that enable software to be transformed from a product into a service.

Companies like Xevo, BMC Software and CyberSource offer supporting technology products as Internet Business Services, another form of SaaS, in areas such as, metering, provisioning, support, management and online payment processing. Software support technologies provide companies, users, or transaction networks with service from centralized data servers, which constantly transfer data via the Internet in both directions to provide services like payment processing or customer relationship management. The data from each action is warehoused and integrated to enhance customer service and product efficacy. Everything from basic applications to complex end-to-end solutions can be delivered in this manner to large and small businesses alike.

Application Delivery Technologies

SaaS services require a variety of enablers, supporting technologies, network providers and hardware providers. Vendors offering application delivery products enable ISVs to readily transform products to services. There are four main ways that the application delivery vendors approach the delivery of applications as a service. These include:

1. *Server-based Computing* (also known as thin-client)
   In the server-based computing model, an application is run on a server, but the user interface is presented through a thin client to the end user. Users can access the output of the user interface via a special client program or within a browser.

2. *Hosted Client Computing* (HCC)
   HCC allows an application to run on the user's desktop, but it is served, or "streamed," from a server. When the user is finished using the application, it is removed without ever having been physically committed to the user's machine. The most efficient HCC technologies segment the application so that only those portions of the application and the data needed to run the application are sent over the network to the user's machine.

3. *Web-based Applications*
   These include the deployment of applications via html. By simplifying user interface logic and separating application data and user data, a web application is developed to be maintained on a server and viewed through a browser.

4. *Java Applications*
   Java allows applications to include a rich mix of interactive features and functions that simple Web - or html - based applications cannot offer.

Customer Focus – Why Buy When They Can Rent?

What value does Software as a Service promise to a prospective customer? In its simplest form, the model allows corporate IT departments to move from application developers to application users, freeing valuable resources to other critical areas. Over-taxed staff can cut back on their software programming efforts in order to play a more strategic role in their company's core mission. The software as a service solution offers the opportunity for companies to select their information technology priorities and then choose from the growing menu of applications now being offered via service providers.

An outsourced service can also provide a superior infrastructure with improved business agility and scalability, and rapid deployment timeframes. In fact, a key selling point of software as a service is the generally much shorter time period required to install and implement new software applications. Given the rapid growth of e-commerce and innovative Internet applications, it is paramount that companies react quickly to take advantage of unique delivery channels and to counteract threats from new "digital" entrants into their industry. Unfortunately, the market is moving so fast and new applications are implemented so rapidly that it is very difficult for the typical company to keep up. With access to a service platform, incumbents and new entrants alike can have access to the latest technology tools integrated with existing infrastructure. Fundamentally, the experience can be an efficient and cost-effective outsourcing process that ultimately gives company management the time and opportunity to focus on the core competencies of their business.

Small and SOHO businesses often have little to no high-end IT infrastructure. These customers are clamoring to improve their competitive positions with technology investments, so that the adoption of various types of planning, productivity and e-business applications can help to achieve new levels of growth

and development. Their problem is they lack the resources to execute the new solution properly. This is a major opportunity moving forward for all types of SaaS companies: helping smaller firms execute technology/Internet-enabled business strategies.

The Customer Value Proposition

An oft-cited SaaS value driver is the reduced Total Cost of Ownership (TCO) for application services.

Cost Comparison Between Having Software In-house and Renting It From an ASP (Based on a real business with 10 users over 3 years)[3]

	In-house	**ASP Rental**
10 Users at $595 month	-----	$214,000
10 Users Running Financial App over 3 years Including Hardware	$60,000	-----
Communications Charges	$100,000	-----
PT Database Administrator	$90,000	-----
Software Support	$30,000	-----
Hardware Support	$45,000	-----
OS Licenses	$5,000	-----
Implementation Services	$65,000	-----
Total	**$395,000**	**$214,000**

The proceeding chart provides an analysis of costs incurred in buying and deploying comparable software on an in-house versus hosted basis. This software rental yielded savings to the customer of 45% over a three-year period. *Software Magazine* (www.softwaremag.com) provides a scenario, below, in which a small, fast-growing technology manufacturer carefully weighed the cost of licensing a nine-module Tier-1 ERP application suite for 50 users versus renting the same application from an ASP. The company cut its IT expenditures by 63% when it decided to outsource the application.

Cost Comparison Between Having Software In-house and Renting It From an ASP (Costs of Nine-Module Tier 1 ERP App for 50 Users calculated over 3 years)[4]

	In-House Deployment			**ASP**
In-house Budget Items	Year 1	Year 2	Year 3	Years 1-3
Hardware (Database and Application Servers)	70,000	70,000	20,000	Included
Software (50 Seat, 9 Module ERP License)	800,000	200,000	500,000	Included
Personnel (DBA & Tech Support)	265,000	265,000	265,000	Included
Yearly Total	1,135,000	485,000	785,000	
Total	+	+	**$2,405,000**	**$900,000**
Average Cost/Month (3 years)	68,800	68,800	68,800	25,000

While these examples show a clear benefit to the utilizing the product via a service model, this may not be the case for all applications. Customers, individual and corporate, must look at their current and future needs as

[3] *Source: Australian Financial Review 4/30/00; Copyright John Fairfax Group Pty. Ltd. Reprinted with permission.*
[4] *Source: Software Magazine, December 1999(http://www.softwaremag.com/archive/1999dec/appsontap.html). Reprinted with permission.*

well as the pricing structure the vendors offer in order to find the best fit for their IT needs. While vendors must take into account all costs, direct and indirect, in providing a true price and value to the customer.

SaaS Licensing and Pricing Models

Today's Software as a Service pricing and licensing models are various and -- at least to some observers -- confusing. Small customers and start-ups love the pay-as-you-go ability that ASPs offer. But established companies, accustomed to paying for software licenses are now faced with the option of renting applications and thereby many choices. Should they enter into a per-user-per-month payment plan? Maybe they should consider transaction-based payment? Annual flat-fee site license? The possible iterations are numerous.

Licensing Models

Software as a Service licensing options include subscription-based, usage-based, transaction-based, value-based and fixed-fee models.

1. *Subscription-Based Model*: Monthly payment is calculated on the software actually used, and includes a commitment as to the actual number of users. Subscriptions are usually written on a per-seat or named user basis.

2. *Usage-Based Model*: Payment is determined by application usage and is typically related to peak or near-peak levels of usage. Payment may be tied to the number of CPUs (customers are charged for every computer that runs the hosted application). It may also be a written for number of concurrent users.

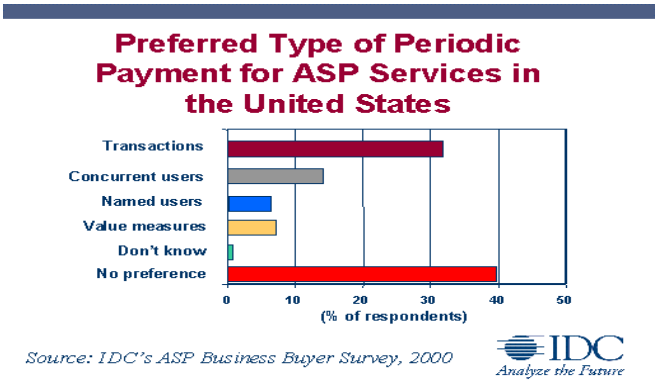
3. *Transaction-Based Model:* ASPs that provide online scheduling and similar products sometimes charge customers for each business transaction: purchasing one introductory relational database class is one service, two classes are two services, and so on.

4. *Value-Based* (a.k.a. Shared Risk or Revenue) Model: Premised on the provision of whatever software is needed to achieve business goals, and payment is linked to the achievement of those goals.

5. *The Fixed-Fee Model*: An emerging option, users generally pay a predetermined monthly fee based on number of users supported, which application modules are rented and service and support levels specified by the customer.

Customer Pricing Preferences Not Met By Most ASPs?

In a recent IDC survey of 267 companies that would consider using an ASP, respondents were asked: "Which of the following types of periodic (e.g., monthly) payment measurements would your organization prefer to use for ASP services?" Their responses reveal that customers prefer models based on transaction and value, but that most ASPs base charges on named users.



In a recent SIIA survey[5] of Certified Software Managers, 50% indicated a preference for a subscription model paid per month or year based on with an unlimited number of users.

Most ASPs seem to have decided on an initial fixed charge for setup and then a monthly fee on a per-user basis, which can be increased as more users are added. The strength of this model is its stability and predictability, particularly compared with the unpredictable and fluctuating costs of applications run in-house. This arrangement is particularly attractive to large customers, who must adhere to strict budget processes, and for whom surprises are never of the "pleasant" variety. So, although customers proclaim a desire for more value or transaction based pricing, the current adopters of application services are more often motivated by characteristics of the per use, per month model.

SaaS Pricing Models

The following chart provides recent examples (11/00) of some price points set by companies offering horizontal business applications as a service.

Provider (Application)	**Set-up Cost**	**Costs/Basis**
Agillion (CustomerPages)		$17.95 - 27.95 / Monthly per user
Corio (PeopleSoft HR)		$750 / Monthly per user
Expercity.com (DesktopStreaming)		$250 / Annually per user
Great Plains (ERP)	$1,500	$4,200 / Annually per user
Interliant (ERP and Collaboration)		$15 - $85 / Monthly per user
Mail.Com (POP or Web Mail Account)		$5 / Monthly per user
Mail.Com (Collaboration and Mail)		$10 - 12 / Monthly per user
Oracle		$895 / Monthly per named user
Pivotal (CustomerHub)		$99 / Monthly per user
Pivotal (IntraHub)		$1,000 / Monthly
Various ASPs (SAP R/3)		$350 - $800 / Monthly per user

Market Development

> The packaging of application services for delivery through online rental computing marks a new shift in the evolution of information technology more significant than the advent of the PC. It will radically change the status quo and become the engine of a new networked economy.
>
> Phil Wainewright, *"Packaged Software Rental: The Net's Killer App"*

The application services market is evolving rapidly. If it were judged solely on number of entrants, this would be one of the great growth stories of the last century. The launch -- within the span of a few years -- of more than 300 entities with the aim of delivering applications over the Internet is nothing short of phenomenal. The growth is compounded by the fact that there is no consensus on the "background" of an ASP -- competitors include software publishers, VARs, ISPs, telecom companies, IT service providers/outsourcers and newly-formed "pure-play" ASPs -- competitors who are also rapidly consolidating.

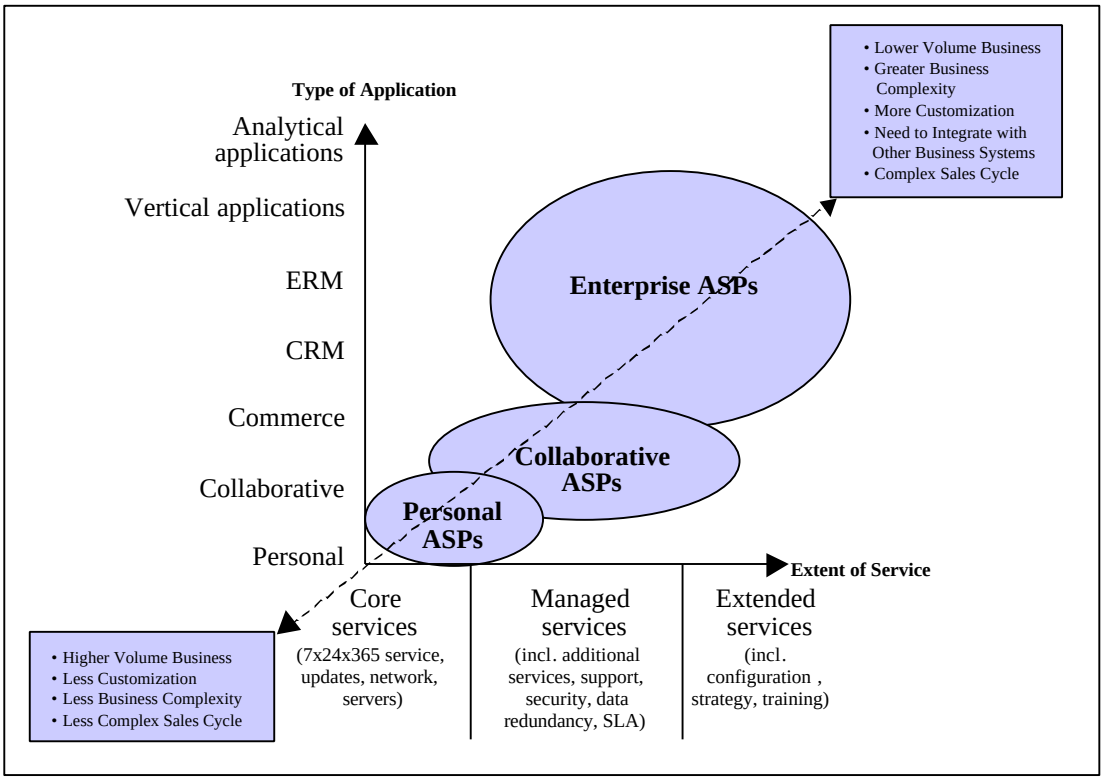
[5] *Source: SIIA* "New SIIA Survey of Corporate IT Managers Reveals Growing Interest in ASP Model" *(http://www.siia.net/sharedcontent/press/2000/11-6-00.html)*

Sizing the Application Services Market

While estimated revenues projected for the ASP market vary dramatically and a wide array of methodologies are used for sizing the market, SIIA believes that the conservative projections for the pure ASP space developed by IDC may prove the most accurate. The SaaS market now comprises about 2 percent of software application sales, but the figure is expected to climb significantly by 2003, according to Forrester Research. Almost all projections agree that the growth rate will run at around 100% a year for the next five years. The major source of discrepancy is in the base—what is considered an ASP in the current market. Since SIIA does not include many of the web hosting, data hosting and other component services as part of the end market, it finds IDC's estimate of close to $8 billion by 2004 to be the most realistic.

Market Segmentation

Categorization of ASP Offerings *(Source: IDC)*



The categorization of the marketplace by IDC, above, provides a good explanation of current offerings. At the high end of the application services market, ASPs offering complex applications and requiring professional services for design, training, implementation, systems integration, and ongoing operations management. ASP efforts at the high end are likely to focus on applications in the following categories: Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Supply Chain Management (SCM), E-Commerce, Data Warehousing, Decision Support Systems (DSS), Workflow & Imaging, as well as select industry-specific solutions. These might feature products and services from such companies as SAP, BAAN, Peoplesoft, i2, Manugistics, Siebel, Oracle, Ariba, and Broadvision, to name a few. The low-end ASPs would focus on personal use, as well as small and medium establishments (SMEs) offering relatively simple applications -- templates that can be configured by users -- and minimal customization.

Projections of Market Development

The following trends are widely expected to impact the development of the SaaS market space in the next year to two years:

- Players are moving between market segments, while integrating new or dropping core competencies. For example, a separation between the ASPs and the infrastructure providers (AIPs) is taking place as the ASPs focus more on application interface, customization and support, leaving infrastructure development and management to the larger AIPs.

- Carrier-aligned ISPs will move into hosting complex front-office and back-office solutions (maybe ERP), either through partnering the providers of managed data centers or by expanding their own internal capabilities.

- Through 2001, ASP winners and losers will be determined by their ability to manage their cost structures, choice of partners, name recognition, repeatable solutions and the ability to execute successfully on required service levels. The result will be significant consolidation in the marketplace.

- Non-telco aligned ISPs will move into the hosted front office and niche application market. In this lower margin "volume business" sector, low maintenance service will be offered. Bundled applications may enable providers to differentiate.

- In the longer term, complete solutions providers (connectivity, data center, software license and integration/support) will emerge since they should be able to achieve significantly higher potential revenues and valuations than a company just forming one part of the value chain.

- Partnerships are becoming critical components of an ASP solution.

- Additional business model dynamics will focus on the struggle between horizontal vs. vertical applications and existing software applications vs. new Internet Business Services applications.

- Enhanced focus on vertical markets and applications – for instance, J.D. Edwards and USinternetworking's recent entries into vertical market apps.

- Connectivity and Internet access are becoming low-margin commodities. The combination of increasing accessibility and declining cost of bandwidth allows a hosted solution delivered over the Net or a thin-client to become both a technologically and financially feasible process. The ASP market is positioned to take advantage of growth in bandwidth worldwide.

Conclusion

Many questions remain on where the Software as Service market is heading and what it will look like when the smoke clears.

We stated in the opening paragraphs of this paper that Software as a Service holds the potential to cause a sea change in the development and delivery of software applications. All signs suggest that the "supply side" of the software as a service market is vibrant and has gained a momentum that will be hard to stem. The massive influx of venture capital, investment in infrastructure, the growth of the ASP value chain and the entry of established ISVs indicate that the model has staying power.

But for those focused on entering the SaaS market, one overarching question surrounds customer acceptance and adoption -- the demand side of the equation -- of software services. It is clear that the large companies and dot-coms have been the key near-term drivers of software service demand. With the shakeout taking place in the dot-com sector, software service demand may face a short-term setback. For the market to continue its rapid growth over the next year or two, large corporate customers will have to embrace the model more completely until the target Small and Medium Size Enterprise market begins to adopt software services.

For the ISV with an existing customer base, the key questions revolve around bringing software services to market with a minimum of disruption to current channels and achieving a maximum additive effect on sales. Key to this success will be partnerships but with the shakeout in the industry in progress, ISVs need to take a critical look at potential partners to assure their long-term viability. There are many questions to be addressed in developing a SaaS strategy, too many to go into here. The next SIIA SaaS document titled "ASP A to Z for ISVs" will delve into the issues that must be addressed - partnerships, privacy, security, data, QoS, IP ownership, etc - and approaches that can adopted, as entrants such as ISVs and system integrators launch software services.

Glossary of Terms

Adapted from *http://www.aspstreet.com/resources/glossary* & *http://www.aspindustry.org/glossary.cfm*.

Application Infrastructure Provider (AIP)	The name given to a firm from whom one sources all the elements necessary to function in a service model. The management of the infrastructure upon which the application is reliant is dealt with by the AIP on behalf of the service provider - in an invisible fashion. It may also take part in the integration of offerings and current infrastructure.
Application Service Provider (ASP)	An ASP deploys, hosts and manages access to a packaged application to multiple parties from a centrally managed facility. The applications are delivered over networks on a subscription basis. This delivery model speeds implementation, minimizes the expenses and risks incurred across the application life cycle, and overcomes the chronic shortage of qualified technical personnel available in-house.
Bandwidth	The number of bits of information that can move through a communications medium in a given amount of time; the capacity of a telecommunications circuit/network to carry voice, data, and video information. Typically measured in Kbps and Mbps. Bandwidth from public networks is typically available to business and residential end-users in increments from 56Kbps to T-3.
Customer Relationship Management (CRM)	Customer relationship management. An integrated information system that is used to plan, schedule and control the presales and post-sales activities in an organization. A full-spectrum CRM application architecture consists of the integrated automation of business processes encompassing customer touch points, including sales (contact management, product configuration), marketing (campaign management, telemarketing, data mining) and customer service (call center, field service).
Electronic Data Interchange (EDI)	A form of electronic communication that allows trading partners to exchange business transaction data in structured formats that can be processed by applications software.
Enterprise Relationship Management (ERM)	Solutions that enable the enterprise to share comprehensive, up-to-date customer, product, competitor and market information; for the end goals of long-term customer satisfaction, increased revenues, and higher profitability.
Enterprise Resource Planning Applications (ERP)	Packages that enable the creation of a single corporate image from disparate, decentralized divisions, enabling users to visualize underlying business processes, reshape these processes and renovate their businesses. ERP modules may be able to interface with an organization's own software with varying degrees of effort, and, depending on the software, ERP modules may be alterable via the vendor's proprietary tools as well as proprietary or standard programming languages. An ERP system can include software for manufacturing, order entry, accounts receivable and payable, general ledger, purchasing, warehousing, transportation and human resources. The major ERP vendors are SAP, PeopleSoft, Oracle, Baan and J.D. Edwards. Lawson Software specializes in back-end processing that integrates with another vendor's manufacturing system.
Hosted Client	HCC allows an application to run on the user's desktop, but it is served, or

Computing	"streamed," from a server. When the user is finished using the application, it is removed without ever having been physically committed to the user's machine. The most efficient HCC technologies segment the application so that only those portions of the application and the data needed to run the application are sent over the network to the user's machine. By centralizing the management of the application, but executing it on the client, it provides the performance and scalability of client-computing with the manageability of sever-side computing.
Hosted Outsourcing	Complete outsourcing of a company's information technology applications and associated hardware systems to an ASP.
Internet Business Services (IBS)	Solutions built to specifically run business processes over the Internet. IBSs offer software services on a "one-to-many" architecture - where the software supports companies, users, or transactions from a centralized data server, which constantly transfers data via the Internet in both directions of the network - to provide services like payment, processing or customer relationship management.
Internet Service Provider (ISP)	Company that provides access for users and businesses to the Internet.
Independent Software Vendor (ISV)	Generally a firm that develops software applications that is not a part of a computer systems manufacturer.
Managed Service Provider (MSP)	A class of service provider that sells subscription services to manage information technology infrastructure and applications over the Internet. (e.g. LoudCloud)
MetaFrame	The world's first Server-based Computing software for Microsoft Windows NT 4.0 Server, Terminal Server Edition multi-user software (co-developed by Citrix).
Network Computing Architecture	A computing architecture in which components are dynamically downloaded from the network onto the client device for execution by the client. The Java programming language is at the core of network computing.
Network Computer (NC)	A "thin" client hardware device that executes applications locally by downloading them from the network. NCs adhere to a specification jointly developed by Sun, IBM, Oracle, Apple and Netscape. They typically run Java applets within a Java browser, or Java applications within the Java Virtual Machine
Network Service Provider (NSP)	A company that provides access to ISPs. Sometimes NSPs also are called backbone providers due to the fact that they provide access to the Internet backbone.
Outsourcing	The transfer of components or large segments of an organization's internal IT infrastructure, staff, processes or applications to an external resource such as an Application Service Provider.
Packaged Software Application	A computer program developed for sale to consumers or businesses generally designed to appeal to more than a single customer. While some tailoring of the program may be possible, it is not intended to be customized for each user or organization.
PRM	Partner relationship management.

Reseller/VAR	An intermediary between software and hardware producers and end users. Resellers frequently "add value" (thus Value-Added Reseller) by performing consulting, system integration and product enhancement.
Server-based Computing	A server-based approach to delivering business-critical applications to end-user devices, whereby an application's logic executes on the server and only the user interface is transmitted across a network to the client. Its benefits include single-point management, universal application access, bandwidth-independent performance, and improved security for business applications.
Single-Point Control	One of the benefits of the ASP model, single-point control helps reduce the total cost of application ownership by enabling widely used applications and data to be deployed, managed and supported at one location. Single-point control enables application installations, updates and additions to be made once, on the server, which are then instantly available to users anywhere.
Thin Client	A low-cost computing device that accesses applications and and/or data from a central server over a network. Categories of thin clients include Windows-Based Terminals (WBT, which comprise the largest segment), X-Terminals, and Network Computers (NC).
Supply Chain Management Application (SCM)	Analyses used to forecast inventory levels and requirements and determine efficient distribution strategies. It analyzes purchase patterns to determine their impact on supply chain effectiveness and efficiency.
Sales Force Automation (SFA)	Methodology that enables sales personnel to concentrate on selling, providing better information exchange in the sales cycle. It includes lead generation, marketing, support and order management/fulfillment functions
Service Level Agreement (SLA)	A contract between the provider and the user that specifies the level of service that is expected during its term. SLAs are used by vendors and customers, as well as internally by IT shops and their end users. They can specify bandwidth availability, response times for routine and ad hoc queries and response time for problem resolution (network down, machine failure, etc.). SLAs can be very general or extremely detailed, including the steps taken in the event of a failure. For example, if the problem persists after 30 minutes, a supervisor is notified; after one hour, the account representative is contacted.
Wireless Application Protocol (WAP)	It is an initiative started by Unwired Planet, Motorola, Nokia and Ericsson to develop a standard for wireless content delivery on the next generation of mobile communication devices.
Extensible Markup Language (XML)	Extensible Markup Language XML is a markup language for documents containing structured information. Structured information contains both content (words, pictures, etc.) and some indication of what role that content plays (for example, content in a section heading has a different meaning from content in a footnote, which means something different than content in a figure caption or content in a database table, etc.). Almost all documents have some structure. A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents.