Peer to Peer

Doing More With Less

09
mar jun sep dec

Peer to Peer June 2009

ASP vs. Saas

Fun with acronyms or a sweeping technological shift?

LARRY PORT ROCKET MATTER

In the dot-com boom era of the late 1990s, we called Internet-based applications "application service providers," or "ASPs" Now we call similar offerings "software as a service" or "SaaS". Why the change in terminology? Are these two things really different, and if so, how?

Both ASP and SaaS providers share a common goal of eliminating IT overhead from business. Hardware, networks, client-server configurations, software licenses and the human resources needed to maintain them are outsourced, and the client organization eliminates those expenses. However, the similarities between ASP and SaaS end there. So what distinguishes these two technologies?

SAAS ORGANIZES CUSTOMER RESOURCES MORE EFFICIENTLY.

Perhaps the biggest difference between ASP and SaaS providers is how they organize computing resources. SaaS customers all share the same environment; servers, code, configurations and database layouts are identical for everyone on the system. This ability to house all customer installations within the same environment is known as "multitenancy." Multitenancy is a term derived from an analogy to tenants in a building. Instead of customers renting space in their own individual buildings (the ASP model), all customers rent space in the same building.

ASP providers, on the other hand, essentially transplanted entire servers and networks from a customer's location to their own data centers. For each customer they supported, they maintained dedicated setups. This business model resulted in data centers filled with servers and configurations of a hopelessly heterogeneous variety.

SAAS ENABLES ECONOMIES OF SCALE.

Since all customers leverage the same code base, database design and machine environments, huge economies of scale can be achieved through SaaS. As more customers, or tenants, come aboard a SaaS solution, costs decrease for the provider.

Upgrades to the system become trivial, since they apply to all customers and no customizations have to be considered. As a result, SaaS companies are able to provide low monthly fees to consumers for access to applications that otherwise can be very expensive.

In contrast, in an ASP environment, each time another customer comes aboard, the provider must set up a dedicated and highly customized installation. Upgrading all customers to newer versions of the software means taking into account the complex setup for each individual customer. This inability to

This article was first published in ILTA's June 2009 issue of *Peer to Peer* titled *"Doing More With Less"* and is reprinted here with permission. For more information about ILTA, visit their website at www.iltanet.org.

control cost and complexity was one of the elements contributing to ASP's failure in the post dot-com era.

SAAS SERVICES ARE DESIGNED SPECIFICALLY FOR INTERNET DELIVERY.

Modern SaaS applications enjoy major advantages over their ASP predecessors in terms of available technology. Massive adoption of broadband Internet access has eliminated slowness and latency that plagued ASP applications. Web standards and solutions emerged to enable desktop-like, or Rich Internet Applications (RIAs). Because of these advances, engineers and designers can build modern SaaS applications from the ground up for Internet use in mind. By leveraging new tools and fast Web access, today's SaaS developers enjoy virtually limitless creative potential.

ASP applications were originally designed as client-server applications and then shoehorned into some form of Web access. In the early days of Internet-based computing, end users had to access their ASP software through crude Web interfaces or remote connectivity software, often over slow dial-up connections. Unfortunately for many of the ASP companies, the technology and infrastructure of the time wasn't quite ready to accommodate full-scale delivery of Web-based applications.

GETTING SASSY WITH SAAS

ASPs were developed with the right intentions, but it took SaaS to get Internet-based application delivery right. Important technological advances in connectivity and developer tools, combined with the implementation of multitenancy allow SaaS firms to thrive where their predecessors could not. **ILTA**

Larry Port is a founding partner and chief software architect for Rocket Matter, an online legal practice management and time and billing application for small to mid-sized law firms. Larry is an active member in his community, speaking on technology issues. He holds an M.S. in Computer Science from New York University and a B.S. in Speech from Northwestern University. He can be reached at larry@rocketmatter.com.

This article was first published in ILTA's June 2009 issue of *Peer to Peer* titled *"Doing More With Less"* and is reprinted here with permission. For more information about ILTA, visit their website at www.iltanet.org.