




DELIVERING STRATEGIC SOLUTIONS ACCA'S 2000 ANNUAL MEETING

Diving into the Unknown? Do ASPs Have Hidden Risks?

Lawyers who embrace ASPs may find themselves in dangerous waters, risking the confidentiality of client data.

By Bob Butler

 SOFTWARE COMPANIES love the ASP business model because it generates recurring income. No need to reinvent yourself every year with a new version of your product. Legal professionals love the ASP model because of easier maintenance and upgrades, plus universal access to data. With both sides in love with the concept, the rush toward the ASP model is unprecedented.

But the unasked question is: "What do we need to be thinking about to keep this love affair from becoming a train wreck?"

Three issues scream for attention: security, availability, and functionality. How secure is the data? How available is the data? Can a browser-based application have all the functionality of the current generation of Windows executable-based applications?

Security

Risking \$50 fraud charges on a credit card is not the same level of risk as having your entire practice database revealed to the public or adversarial party, or to have that data lost or unavailable.

ASPs present unique issues for attorneys. For example, has your professional liability carrier taken a position re: exposure on ASPs? What are the liabilities if confidential information gets revealed or lost? Has your bar association passed any policies or created any standards? Without a standard or much precedent for what is reasonable and prudent care regarding online data, what is your disciplinary exposure for revealing or losing critical client data?

Beyond special aspects of the legal profession, are the ASP risks different from the security risks we take every day? Experts will tell you that nothing is really secure, it is always a question of balancing risk

versus reward issues. They offer variations of the old argument, "While this burglar alarm will not deter the criminal determined to get into your house, hopefully it will make your house less attractive to the average criminal than your neighbor's house without the burglar alarm." Does this argument really apply to the Internet and the ASP?

In addition to the complex technical aspects of Internet security, there are at least two fundamental reasons that Internet security risks are greater than traditional security risks:

1. Ease of transferring security-breaching knowledge
2. Lack of physical site risk to the person breaching security.

Good safe crackers, expert in bypassing alarms, etc. take many years to learn their craft. In the Internet world, once security is breached, the knowledge can be instantly transferred to literally thousands of others. The security threat never has to be at the site, where passers-by, nosy neighbors, police on patrol, etc. may discover them. These two elements alone make Internet security risks potentially higher than other security risks, even with everything else being equal. These two fundamental difference essentially mean that the house with the burglar alarm can be as easy to enter as the house without an alarm.

Therefore, one of the most fundamental tenets of security may not apply to the ASP. The risk reward balance may be shifted toward risk with ASPs.

Security is only as strong as its weakest link. The latest bastion of Internet security, asynchronous encryption (public key/private key encryption) is very secure from the point of view of preventing decryption of Internet communications. But, there are real issues about access to the private keys.

The private keys are just files on your hard drive that are password protected. The whole argument for asynchronous encryption is that passwords provide a much lower level of security. These private key files can be accessed on the hard drive and compromised. This encryption approach may be like a super-secure locking door, but with the key kept under the doormat.

The new secure IBM PC desktops, where the private key is built into the motherboard and not on the hard drive or PC bus (see LTN, March 2000, page 1) is a step in the right direction. Biometric devices, such as thumbprint keyboard and security cards, typically connect into the PC bus. Like the hard drive, anything on the PC bus is vulnerable. Is your ASP instructing you on what hardware and software to buy and how to use it to keep your data secure?

The reality is that very few individuals have the knowledge to breach these systems. But with the Internet it only takes one person with the capability to breach security, then broadcast that capability to everyone else.

There have been very few reported breaches of ASP security. But there is very little ASP data on the Internet right now. As the amount of ASP data increases, so will the desire to access this data, and so will the number of security breaches.

Next, do you know where your ASP hosted data actually is? Who actually has the physical care, custody, and control of the data? Most ASPs will contract with larger ISP or telephone companies to provide the server hosting and Internet access. Your data may not actually be stored in the offices of the ASP.

Even if you do have security and access provisions in your contract with the ASP, can you be sure those transfer to the subcontractor. What security checks have been run on ASP or subcontractor personnel? What if one of the technicians at the ASP or subcontractor has a friend who works for the tabloids? Better yet, will the ASP also make server technology available to you so you can host the data from your own offices?

Another issue that few ASPs have considered is security within the database once you have granted controlled access to the database from the outside. Does the ASP application have complete control and audit trail right down to the record level? Field level? Can you feel comfortable letting your clients, outside counsel (or whoever) wander freely throughout your database without intra-database security controls and an audit trail?

While most ASPs have thought about what happens when the Internet is down, very slow, or your ASP is under a denial-of-service attack, etc., has your ASP invested in technology to solve those problems? Does the application have a local copy of the database, perhaps with IP-based synchronization running in background with the ASP database, so you still have access to reasonably current data if the ASP database is not immediately available?

In addition to Internet reliability, there are other data availability issues. For example, how available is your data when you want to take the data elsewhere? Who owns the data if you have a dispute, or don't pay your bill, or the ASP goes broke? What if you want to transfer to another ASP?

Will you have the legal right to transfer your data? Does the ASP have the technology to let you create an export file of your data in an industry standard format?

Functionality

The ASP model has two options for developing the software you will actually use day-to-day:

1. Make a typical Windows application browser-capable

- (executable-based application or thick client), or
2. Make a browser application-capable (browser-based application or thin client).

An executable-based application will be faster and have stronger features than a browser-based application. Customers have been known to complain about a procedure taking two seconds instead of one second or two mouse clicks instead of one click. Will customers stay satisfied with browser-based applications that do not include all the performance and feature refinements they are used to with executable-based applications.

Yet, the universal availability and ease of access of the browser-based applications has an undeniably strong appeal. In the end, it's a safe prediction that both application types will find their place. Application-based for the power users and everyone else back at the office, and browser-based for the road warriors and outside parties. The question is: "Will your ASP have both application types available?"

Right now there are two business groups rushing to the ASP model: Software developers with experience developing strong applications, but less Internet experience; and Internet companies with extensive Internet experience, but very little experience with developing full-featured applications.

The winners on the business side will be those that can quickly acquire the experience and expertise of both groups while maintaining a strong focus on database security and availability.

The winners on the customer side will be those who ask all the right questions.

[Bob Butler](#) is technical director and co-founder of [Time Matters Software](#).

This material is protected by copyright. Copyright © 2000 various authors and the American Corporate Counsel Association (ACCA).