



Luis Segredo

President

M-Tech

POINT

The Shift Has Begun

SaaS solutions are already proving that fetching data or logic on demand is a more efficient and safe way to make software solutions communicate.

When you go to Google and enter keywords, you're not searching against a local copy of Google. When you go to MapQuest, you don't query a local copy of all of the nation's maps. No, all of the data and logic resides out on the Web. When you want to access these resources programmatically, each has APIs that can be called in real time. Yet, when there's talk of making applications communicate in hospitality, we don't think of fetching information from other applications in real time. Instead, the conversation all too often revolves around replicating data between systems. Guest profile data gets copied to reservations. Reservation details are copied to a myriad of systems, and the chain continues.

By copying the data from system to system, challenges are introduced and opportunities for better integration overlooked. First, copying data is not efficient. Whether it is to be used or not, it has to travel from system to system and be replicated. Once it is copied to two or more places, any changes need to be synchronized. Each system needs to also create user interfaces and logic to make use of this data. Then, there is the issue of security. Because data is stored in more systems, the opportunity for a security breach increases. When these hurdles are overcome, new versions of software and data models allow the process to start over. This method of interfacing is archaic.

The reason we see so much interfacing happening this way is because of the nature of applications in the hotel industry. Many applications are still site based. In order to break the cycle of copying data from system to system, each vendor needs to feel confident that the information that they require will be available in a timely fashion when its application requires it. The examples given above, Google and Mapquest, are provided over the Internet, so the onus of performance is placed on the providers. The providers ensure that they have sufficient bandwidth and hardware to handle requests in a timely manner. If something goes wrong, then it is their responsibility. When an application resides on property, then the software provider has to rely upon the customer's network and hardware, so the ability to ensure a request is handled in a timely fashion is hampered. The result is many or most interfaces between applications end up replicating and synchronizing data and/or logic to get around this. They communicate as events occur with the hope that each system will have the data when they need it. These interfaces are heavier because of the resulting data synchronization and storage issue and therefore costlier to develop and maintain.

It does not have to be this way. More and more software providers and users are turning to hosted models or software as a service (SaaS). This model is proving to be more cost effective and yields higher reliability. As more users go SaaS, then interfaces can be on demand, similar to Google or MapQuest. When information from one solution is needed by another, it is fetched in real time. You're probably wondering how vertical market software can be compared to giants like Google in providing availability. Well, if you prefer, I can use an example that is closer to home, salesforce.com.

Salesforce.com has an extensive API that allows other applications to communicate with it in real time. External vendors are encouraged to write applications that interact with salesforce.com, and dozens upon dozens of applications have done so already. An example that is even closer to our industry is HotSOS. Our company exposes not just data but also functionality that can be called by other applications on demand. Today, external applications, like PMSs and others, fetch raw data or even screens and logic from our servers to display within their applications. This gives a level of integration that is unparalleled in functionality or ease of development. The last such integration project was accomplished in a matter of days and required only a couple of e-mailed questions.

From an industry support standpoint, HTNG is promoting Web services for in demand integration as one of its supported methods. Because of the challenges mentioned above, it is not the only method, but HTNG recognizes the opportunity to reduce the duplication of data wherever possible. HTNG members developing both site-based and SaaS solutions, like PMSs and others, are now espousing Web services that expose data or logic as needed. The shift has begun.

From the standpoint of security, housing data in one place is preferred. By promoting the use of data on demand, the energy to secure it is focused on one place. It is easier to secure a house if it only has one door.

SaaS solutions are already proving that fetching data or logic on demand is a more efficient and safe way to make software solutions communicate. The proliferation of Web services is also working its way into site-based systems. When, given the additional challenges, site-based systems are joining SaaS solutions in taking this approach, it is testament to the future. Hopefully, it won't be long before this is how all applications communicate.