SOA AND SAAS:

GETTING THE BEST OF BOTH WORLDS

How Flexible Enterprise Architecture and

Software-as-a-Service can Deliver Value

David Dahl, Chief Technology Officer

SpringCM

SOA and SaaS: Getting the Best of Both Worlds

Table of Contents

1 | Overview

New uses of existing technologies are providing companies a degree of flexibility never before seen. Today more than ever, CIOs possess the tools to keep their organizations nimble. By utilizing Software as a Service and Service Oriented Architectures, CIOs can easily react, even lead, this new phase of business transformation.

To start, let's clarify the definitions of these popular terms:

Service Oriented Architectures (SOA) are not products; they are an architectural style based on long-standing technologies and standards. SOA provides a flexible, loosely-coupled way of implementing a system, focusing only on *what* a service does and ignoring the *how* or *where.*

Software as a Service (SaaS) is a software delivery model, wherein a common code base is maintained in a multi-tenant instance. Customer configurations and connections sit in a layer above the core code and the complete operation – development, deployment/hosting, maintenance and upgrades – is provided by the software developer as a service and often priced as a subscription. SaaS is also referred to as "on demand," because capacity and seats can be added or subtracted, as the user requires, without additional procurement or deployment efforts.

The risk with early SaaS applications was that a department level buyer would identify and procure a solution that met a business need, was inexpensive and quick to deploy, but inadvertently created another information silo – this time "in the cloud."

> **By utilizing Software as a Service and Service Oriented Architectures, CIOs can easily react, even lead, this new phase of business transformation.**

Today, by using Web services to connect applications, whether inside the enterprise or off premises, CIOs achieve the best of both worlds. Software as a Service and Service Oriented Architecture can work together to give CIOs a new degree of cost-effective technical flexibility.

2 | Historical Perspective

Initially, information systems were deployed based on available technologies, leading to the current agglomeration of solutions at every level. Additionally, computing power was locked in mainframes. Access to this monolithic resource was through dumb terminals, and valuable processing power was divvied up by time-sharing between users or jobs. Over time, Moore's Law transformed this landscape by providing more cost-efficient means of accessing computing power. Enter the time of mini-computers. Separate lines-of-business eventually realized they were able to cost-effectively manage their own computing needs.

The next phase emerged when micro-processors and micro-computers began sprouting up on every desk. Similarly, smaller more cost effective servers began showing up in the line-of-business Information Systems (IS) departments. While cost-effective systems enabled separate departments to provide their own point solutions, they also created hardened silos of information across the organization. Marketing, Accounting, Human Resources, and other departments maintained separate systems that didn't communicate effectively . . . if at all.

In order to control and predict costs, companies invested heavily to develop custom applications to manage procurement, workflow, and other production systems. There was a need to develop software solutions that could provide an enterprise-wide platform for managing all the various islands of information across a business. Enter the age of Enterprise Resource Planning (ERP). ERP systems ushered in a new era of focusing on technology to drive efficiencies across an organization.

2.1 | When ERP Roamed the Earth

ERP vendors began offering a single enterprise-wide application designed to replace the siloed systems, which provided coordination and visibility. Many companies bought into the notion that the only way to control this sprawl was through custom development or ERP investment. Their desire was to internalize, control and interlink business processes. The shift to ERP was always costly and often ineffective.

Interoperability of systems and interoperability of processes was not effectively realized because of inherent complexities including process and change management barriers, and inadequate infrastructure to name a few.

While there are many success stories to be found, most corporations learned a hard lesson by purchasing large-scale ERP applications, committing themselves to lengthy and expensive implementations. In the end they were unimpressed with the system's ability to meet changing business needs. The need for more effective systems and process communication was not met.

2.2 | Evolution and Devolution of the ASP

In the late 1990s service providers attempted to take an element of pain out of the equation – the management of the data center and the hosting of applications. Widely available connectivity, disaster protection, and robust technology enabled hosting applications off premises. However, the model did not really result in lower costs or substantial efficiencies – just a shift of costs onto the provider.

Fundamentally, outsourcing software wasn't a bad idea. It actually allowed companies to focus on critical systems and processes. However ASP capabilities were based on multiple installs of traditional single-tenant software. This architecture was not scalable and was difficult to manage. The ASPs inability to extract costs out of the economic system lead to their near extinction.

The ASP model fell apart largely because of three issues 1) mismanaged expectations of customers on customization, 2) inability to get any development or maintenance leverage a single-tenancy model, and 3) the overall economics of the ASP alternative.

2.3 | Interoperability is Born

The era of ERP was important because it attempted to force communication between legacy systems. During that time, technologies evolved that became known as "open" technologies for their well-documented interfaces and ability to easily interoperate. Service Oriented Architectures, or SOA, utilize a subset of these technologies.

The technologies that support SOA (primarily Web Services, CORBA, or RPC) are not new in and of themselves but, coupled with a new implementation paradigm, enable a new class of enterprise-level systems. Today, SOA's proven results are commanding respect within Enterprise Architecture groups and strategic planning circles.

Implementation of SOA-based architecture allows new and old systems to interoperate equally. The independent systems don't know or care what platform a peer-system is running, or even where it is physically situated.

Technologies such as Web servers, distributed databases, middle ware and messaging services, have come together to enable flexible systems, often communicating over Web services as their interface, and TCP/IP as their backbone. SOA has clearly emerged as the set of technologies on which to base systems prone to change.

2.4 | The Browser and Broadband Both Grow Up

While SOA was transforming mid- and back-office systems, interesting maturation was taking place on the Web. New development approaches changed browser based applications into tools that were pleasing to the eye and the brain – clean cut usability from Flickr, del.icio.us, and Google are just a few examples.

Secondarily, wide adoption of Internet technologies in the consumer market meant that the underlying technologies had to mature – enabling security, performance and scalability to be delivered to millions of users at once.

Ultimately, the realization of widespread broadband access, confluence of technologies, infrastructure and experience, created the Web we know today – a place of true enterprise-level, business-to-business commerce. And a new class of Web-based applications for business, sometimes referred to as Enterprise 2.0, emerged.

2.5 | Emergence of SaaS – The Right Twist on an Old Idea

Software as a Service might be called "ASP meets Amazon." A business application is managed and maintained off premises, like an ASP, but with consumer level usability, an ecosystem of participants, and customization or configuration data stored above the software's core logic, like Amazon.

Salesforce.com changed the way corporations looked at putting critical business information on the Web. With over 30,000 customers, Salesforce.com helped in large part to bring Software as a Service into the corporate mainstream. Now, companies of all shapes and sizes are embracing the SaaS model, leveraging Web-technologies and the pay-per-use model. Companies such as Salesforce.com and SpringCM, are examples of native SOA-based systems providing a Web 2.0 look-and-feel, and delivering their services over the new SaaS model.

Initially, SaaS providers did not provide much in the way of system-to-system interconnectivity (i.e. application programmer interfaces – APIs): they were offered as pure standalone solutions. Now, the convergence of SaaS, SOA, and Web 2.0 give today's CIO the ability to interconnect components situated in various data centers, built on various technologies, and operated by various companies.

These new systems, which are quickly integrated, are referred to as Enterprise Mashups. Enterprise Mashups sit squarely in the world that Web 2.0, SOA, SaaS, and Business Process Management have created.

3 | Realization of SOA and SaaS

Traditionally CIOs are challenged with keeping their systems and solutions relevant to changing business needs. More and more, as technologies and methodologies mature, technical leaders find themselves better equipped to not only support the business, but also identify and realize new solutions contributing to corporate strategy. SOA and SaaS are significant contributors to this new era of technical leadership.

SOA and SaaS are separate, but intertwined concepts. The point at which these become intertwined, is when considering system-level integration of a SaaS partner. Even though challenges exist, the benefits of creating a SOA enterprise, and extending it to SaaS partners, can be persuasive.

3.1 | Benefit 1: Breaking Down Technology Silos

Years ago, point solutions created within individual lines-of-business worked to a degree, but they didn't interoperate. Systems, data, and processes became silos within a corporation and therefore were completely not extendable. Monolithic systems (such as ERP) attempted to subsume the silos and create gigantic enterprise-wide platforms – they worked, but rarely. With SOA, a straight-forward, cost-effective, and incremental approach to connecting disparate systems, data stores, and processes is available.

Both legacy systems and new systems need to have the appropriate connectors implemented – simply put, they need to be opened up. Once they're enabled for interoperability they need a common forum in which to communicate. This forum is called the "service bus"[1] and supports service discovery, interaction, and usage of services on *any* system in *any* location.

Essentially, SOA technologies provide a platform to plug in any silo, making its capabilities and information available to any system plugged into that same platform. This model can extend outside the enterprise to enable system-level SaaS integration, further enabling the enterprise platform.

3.2 | Benefit 2: Provide Technical Flexibility

The SOA-based platform described previously is designed to be very loosely coupled, which means that services have a clear awareness of each other, yet remain completely independent. Applications, or even component capabilities, are easily disassembled or reassembled based on loose coupling. Component applications benefit the organization because it makes changing the service provider or user much easier. For example:

- Changing physical location of provider or consumer services
- Upgrading, maintaining, supporting services, underlying software or hardware
- Real-time scaling of additional services
- Abstraction from implementation details

The services, by definition, provide a contract based on descriptive metadata. As long as a service respects the contract it publishes, the implementation details, physical location, and type of architecture are irrelevant to the consumer.

3.3 | Benefit 3: Drive Corporate Focus

Integrating an external SaaS provider allows companies to focus on their core business. Implementing a new system or solution can lead to obvious technical distraction on a budget and resources[2].

- Software development and/or implementation
- Application monitoring, service, and upgrading
- Outsourcing repeatable processes that can be easily defined

SaaS-based solutions allow a company to focus its financial and personnel resources on truly core/critical systems and processes: those that are overly complex, or require dedicated in-house ownership and attention. Repeatable processes and capabilities that can be outsourced should be outsourced – cutting through technology clutter.

---

[1] Service Bus' are discussed in more detail in Section 4: Evaluating SOA Technologies

[2] See Section 3.4: Reduced TCO for more on SaaS cost reductions

3.4 | Benefit 4: Reduced TCO

An internal SOA implementation can reduce overall system Total Cost of Ownership because of its flexibility over proprietary middleware which tends to be costly, unwieldy, and static-binding.

Costs can also be reduced through connecting SOA to a SaaS provider. Initially, IT can easily open point systems to a SaaS provider using a simple secure gateway. This way, you can quickly realize SOA-like benefits by learning from a SaaS provider, grow organically into an organization, and phase in systems to manage risk.

- The first SaaS interaction can be used as a low-cost experiment in the development, deployment, and support of externally available services.
- A second phase could be undertaken to further grow service connectivity within an organization by replacing the single service solution with an ESB open to all providers.
- Finally, you can connect other existing services to the bus for consumption.

Two of the main tenets of the SaaS model are: 1) pay as you go, and 2) fast and easy provisioning. SaaS solutions provide excellent capabilities at a fraction of the initial cost of a traditional ISV solution. Furthermore, if a SaaS integration does not realize the expected benefit, the opportunity cost is low.

Specifically, SOA reduces costs by:

- Decreased integration expense
- Increased reuse
- Managing risk

And SaaS reduces costs by:

- Decreased expense leveraging SaaS economies of multi-tenancy
- Low (or Zero) ongoing costs for software, hardware, or FTE
- Vendor expertise, scalability, and ownership

3.5 | Considerations and Risks

Implementing an internal SOA-based Enterprise Architecture is not a trivial task. Technical leaders need to consider the following and more:

- **User Requirements:** Consideration needs to start with determining exactly what services users need.
- **Security:** whether internal or external services, appropriate authentication and authorization is required.
- **Service Granularity:** Defining what business process a service consumer requires, then determining the atomic sub-services, and their appropriateness.
- **Quality Assurance:** QA of the development, unit testing, and acceptance testing of service interfaces can bring brand-new challenges to a QA team.
- **Monitoring**: New tools and standards need to be applied for appropriately tracking and alerting on the health of internal or external services.

4 | Evaluating Considerations

SOA and SaaS are separate but intertwined concepts. The section below addresses some of the topics to consider when entertaining a SOA and/or SaaS implementation.

4.1 | Evaluating SOA Technologies

As mentioned previously, a flexible SOA solution can bring about many cost, market-speed, and support benefits. But a full service-oriented architecture is not typically required to connect at the system-level with a SaaS provider. Simple connectivity could be realized by minimal web-service integration.

However, to realize the full benefit of taking an Enterprise down the full SOA-path, technology leaders must consider the wealth of ISV technology in the marketplace today.

SOA is based on fundamental, well-understood, and time-tested concepts. Therefore third-party software is by no means required. But the software marketplace has come to the rescue nonetheless with solutions that *can* ease implementation, support and maintenance, through a commonly understood product called an Enterprise Service Bus (ESB). SOA technology evaluations should be focused on the ESB.

According to Forrester, the ESB is "*Infrastructure software that makes reusable business services widely available to users, applications, business processes, and other services.*"[3] So, clearly, for a robust SOA implementation: um, get on this bus. They'll bring you tools and frameworks to specifically support Transaction Management, Security, and Request Routing.

There are two unsurprising large-scale vendors worth looking at: BEA's AquaLogic, and IBM's Websphere platform (consisting of WebSphere Enterprise Service Bus, WebSphere Message Broker, and WebSphere Process Server). Additionally, long-time integration enabler IONA has their own robust product called Artix that should be weighed equally with the gorilla-products.

4.2 | Evaluating SaaS Providers

SaaS providers come in many different shapes and colors with varying levels of integration capabilities. There are companies providing solutions at all levels in the organization.

When evaluating potential providers, CIOs need to consider the following topics[4]:

- Integration Options: Many providers provide only user-interface customizations, some not even that. Other providers such as Salesforce.com provide system-to-system connectivity into their SOA stack to enable intimate event-driven processing. SaaS vendors such as SpringCM provide both of these integration capabilities.
- Data Integrity and Control: Technical Leaders need to understand where their data goes once it passes through their firewall. Who within the SaaS organization can access this data and what other 3rd parties may be involved with data backups and storage. Data Backup and Retention Policies should be readily available.
- Service Levels and Planned Downtime: SLAs are still not prevalent in the SaaS world. But they're coming. As the industry matures so will customer expectations. Regardless, potential SaaS vendors should still be able to quote mean service levels, service commitments, and policies for taking planned downtime.

While the framework described here is important in due-diligence for any SaaS evaluation, fundamentally, CIOs need to weigh the breadth and depth of the evaluation to correspond with the critical nature of the data/process being outsourced.

---

[3] *The Forrester Wave™: Enterprise Service Bus, Q2 2006; Vollmer and Gilpin; June 30, 2006*

[4] *See the SpringCM whitepaper "P.R.A.I.S.E: What to Look for in Evaluating a SaaS Provider" www.springcm.com*

5 | Real World Examples of SOA and SaaS Working Together

Technology groups can use Web services and SOA to get more done for their internal customers. At SpringCM, our customers use Web services to execute document process automation using SaaS and SOA. In all cases, the results have yielded a very high ROI.

Through SpringCM, each customer has access to a complete ECM solution for a department or process need. Users access SpringCM directly through a Web-based interface, other through other applications in which they may be working, critical points of information exchange with legacy systems, or other applications that ensure information flow.   The following examples illustrate several SpringCM implementations, and show how SOA and SAAS can work together to automate document processes, and solve problems for operational constituents, for a significant ROI.

5.1 | Making a Call on a SpringCM Web Through External UI – Proof of Delivery Automation for Revenue Acceleration

Priority Solutions Inc., a global logistics solutions provider in the pharmaceutical/ healthcare industry, uses SpringCM to process compliance and delivery documentation. The customer's proof-of-delivery (POD) information is transferred to the SpringCM system, indexed and processed for optimized storage, and is then available for search and retrieval. Priority Solutions' employees access this POD information through their own intranet built upon the SpringCM Web services. Priority Solutions' support personnel can easily execute a search through their internal portal, view the results, and download the specific POD, all from within their secure network. The SpringCM solution passes all queries, results, and documents over a secure Web-service channel. Utilizing SpringCM, Priority Solutions has increased the accuracy in support and administration of shipping documentation while decreasing the overall cost to develop, support, and maintain a similar solution.

5.2 | Triggering Actions in an Internal System Through SpringCM – Insurance Broker License Renewal Notification for Penalty Avoidance

A large health insurance provider has integrated with SpringCM to manage the license renewal of its field sales staff. Approximately 2,000 brokers upload license documentation into SpringCM.   Once documents come in, they are tagged with expiration dates – this is the repository of record for the broker management team. In the event that a document expires, it triggers a report that is shipped via Web services to the customer's system, which reacts by alerting accounting and other systems that a broker is out of compliance. As a result, the company avoids significant insurance commission penalties, and is much more efficient.

5.3 | Web Services to Connect ERP Systems for Data Transformation – Chemical Distribution Content Delivery for Lowered Costs

CHEMCENTRAL, a global industrial chemical distributor, has integrated with SpringCM to more efficiently manage the online and offline creation and delivery of documentation, such as price-letters and invoices pulled in from the customers' JD Edwards system. The JD Edwards system generates descriptive data which is uploaded to SpringCM. SpringCM transforms the system data into customer specific documents and delivers it to the right person over the right channel. Managing the CHEMCENTRAL address book is just as seamless. The delivery preferences within the address book give CHEMCENTRAL the ability to send content over email, fax, or USPS – all through seamless system-level connectivity. As a result, costs associated with creating, delivering, and managing this information have dropped dramatically.

5.4 | Accessing Documents in SpringCM From Within a Record in an Internal Financial Application

SpringCM has integrated to a common accounting package to allow access to purchase orders and invoice supporting documentation in the context of a financial record. Supporting documents are loaded and indexed in SpringCM, routed for approval, and then placed into a synch folder. When the accounting package is activated and online, the system connects to SpringCM and synchronizes the data fields for anytime retrieval of content from the SpringCM repository.

In conclusion, these real world examples show how the intersection of Software as a Service and Service Oriented Architectures have enabled cost reductions while increasing overall system and process flexibility. SaaS and SOA give today's CIO a new set of capabilities for solving business challenges. Initial SaaS/SOA rollout can be controlled and implemented in phases. Once fully embraced, these technologies allow a technical organization to further leverage legacy systems from the past, while future-proofing an organization.

6 | Summary

SaaS providers are front-runners in providing an architecture open to SOA, allowing businesses to easily extend their systems and cost-effectively streamline business processes.

CIOs who internally embrace SOA-based enterprise, and connect with external SaaS-enabled service providers, can help focus their internal resources on core competencies, and provide the most cost-effective solution, while further leveraging the sunk-cost of legacy applications and future-proofing new applications.

The combination of SOA and SaaS will better enable CIOs to provide the required internal and external services in a changing business environment and equip them with the tools to provide strategic capabilities and flexibility to help drive corporate strategy.

References

About the Author

David Dahl is Chief Technology Officer of SpringCM.   David comes to SpringCM after serving as VP of Technology Development with Orbitz. He has extensive experience in developing dynamic consumer service and private label booking systems for major airlines. At Orbitz, he launched the web services and private label operation for NWA.com. David has held high-level positions at Zefer Corp, Andersen Consulting, and Nortel Networks. His client base included major projects for Wells Fargo, State Farm Insurance, Ameritech and Allstate. David is an avid Black and White photographer, and cyclist. BS, Purdue University. MBA, Loyola University, Chicago.

E-mail David at **ddahl@springcm.com**

About SpringCM

SpringCM is the leader in on-demand document and content management, helping companies of all sizes, both domestically and internationally, automate their document processes, improve communication, and collaborate, both inside and outside their company walls. SpringCM meets the real need of businesses to manage the explosion of documents and other content types – from capture, storage and search, through collaboration, workflow, and document process automation. Today, customers such as Avon Products, General Electric, and Cox Communications are utilizing SpringCM's unique solution to solve their document management needs. SpringCM maintains offices in California and Chicago. The company is backed by Foundation Capital of Menlo Park, CA. For more information visit **www.springcm.com** or call us at 312.881.2026.

[Photography in "SOA and SaaS: Getting the Best of Both Worlds": courtesy of David Dahl]