Evolution of SOA

How today’s leading organizations develop SOA with lower upfront investment and risk

Abstract: Since it first emerged more than a decade ago, Service Oriented Architecture (SOA) has been both widely praised as a modern and agile approach to software development and infrastructure architecture and dismissed as a colossal waste of time and money. Far from being dead, however, service oriented architecture is in fact more relevant than ever. The key challenge for companies is to determine how to best enable SOA within their organizations. Learn how a “bottom-up” approach enables organizations to implement SOA with lower upfront investment and risk.
1 Service Oriented Architecture Overview

Since it first emerged more than a decade ago, Service Oriented Architecture (SOA) has been both widely praised as a modern and agile approach to software development and infrastructure architecture and dismissed as a colossal waste of time and money. Missed expectations for many SOA projects led one analyst in 2009 to declare, “SOA is dead.” Far from being dead, however, service oriented architecture is in fact more relevant than ever. The failure of SOA projects has less to do with the fundamental nature of SOA as a set of design principles and methodologies and everything to do with how vendors have architected their SOA offerings and how enterprises have implemented them. It is clear a new approach is required to deliver on the initial promise of SOA without the pitfalls that have tarnished its reputation.

Enterprises continue to face the same infrastructure challenges that SOA was designed to solve in the early 2000s, including increasingly complex architectures and code redundancies. Moreover, given the unpredictable nature and challenges of macroeconomic conditions, organizations need to rein in IT costs to meet budget constraints and increase their agility to stay competitive in an increasingly cutthroat environment. The key challenge for companies has been, and continues to be, how to best enable SOA within their organizations. Core to this challenge is choosing the right approach to deliver an immediate impact and ROI without major disruptions to existing development processes.

Many forward-looking organizations have adopted approaches to SOA that delivered exactly this. Drawing from the lessons of these organizations, this whitepaper provides an overview of SOA uses and benefits, an analysis of different approaches to enabling SOA, contrasting between “top-down” and “bottom-up” solutions, and best practices for selecting an SOA solution.

2 Why SOA? Uses & Benefits

Fundamentally, service oriented architecture (SOA) is a model of infrastructure architecture and an approach to internal application development. Before SOA emerged in the early 2000s, enterprise infrastructures consisted of multiple applications, typically developed in-house to provide a new business service or automate a particular business process. Often, applications for related business processes contained duplicate functionalities, with the same code existing in several internal programs. For example, if multiple programs required credit check information, each of those programs would duplicate the code needed to perform the credit check. These additional codebases resulted in multiple inefficiencies. Code was poorly reused, leading to wasted effort and money spent during development. As infrastructures became more complex over time, it became increasingly difficult for developers to maintain and support these applications. Finally, businesses lost agility. If the credit check process needed to be changed, this required multiple developers to update multiple applications, slowing down the entire modification process. Overall, these overly complex and poorly designed applications had substantial impact on both the top and bottom lines of the organization.

The breakthrough of SOA was to design infrastructure architecture around services rather than entire applications. In such an architecture, the emphasis is on creating components called services, which are
small, discrete units of software that provide a specific functionality and can be reused in every application. In an SOA model, developers thus create new applications by orchestrating a collection of services instead of building out an entire software program, eliminating code redundancies across multiple applications. For instance, in SOA a simple bank loan application would be a composite of credit status check services, interest rate services, customer data services, and more.

In short, SOA breaks down the islands of business logic and data that are scattered across multiple, disparate applications. It enables interoperability between systems through integration, making it easier and faster to automate business processes.

The benefits of SOA are plenty. By improving the agility of IT systems and business processes, enterprises can better respond to changes in the market and innovate new products to stay competitive. At the same time, they can reduce the bloat and complexity inherent in legacy systems, increase developer productivity by making software design more intuitive, and lower IT costs associated with maintenance and upgrades. It’s no wonder, then, that so many companies rushed to implement SOA projects when the concept first emerged.

3 Best practices for selecting an SOA solution

There are several different approaches to implementing SOA and selecting the right solution for mission critical environments. It is a common mistake, however, for enterprises to jump into a lengthy and costly IT project without fully evaluating overall business objectives, technical considerations, and project requirements. When selecting an SOA solution, top questions to ask include:

<table>
<thead>
<tr>
<th>Business objectives</th>
<th>Technical considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>• How is my industry changing and what short-term and long-term needs will I need to address? How can I efficiently address current needs while establishing a framework for future growth?</td>
<td>• Does my SOA solution provide support for all of the technologies in my IT environment? Does it support emerging technologies such as cloud and SaaS solutions?</td>
</tr>
<tr>
<td>• What new business services do I plan to offer? Does my current infrastructure support agile growth and provide enough flexibility to quickly pivot to changing business needs?</td>
<td>• Can my SOA solution be used with application servers, development tools, etc. from different vendors or will it force vendor lock-in?</td>
</tr>
<tr>
<td>• Which SOA solution provides the best ROI?</td>
<td></td>
</tr>
</tbody>
</table>
4 “Top-Down” SOA

When enterprises first began to adopt SOA in large numbers, many opted for a “top-down” approach. Often this entailed launching a single organization-wide SOA initiative and selecting a highly closed proprietary SOA stack from a big vendor such as IBM or Oracle. Once purchased, the company would begin a planned rollout process, often with the help of consultants. Once complete, the company’s development team would then need to learn and use the product to re-architect all existing systems as well as design new applications according to SOA principles. These developers would have to throw out their existing tools, processes, and skillsets and be heavily retrained on the new solution.

Comprised of multiple products, including application servers, enterprise service buses, orchestration engines, management tools, and development tools, proprietary SOA stacks provide a relatively robust and reliable platform for service orchestration and integration. Nonetheless, using an SOA stack to implement a top-down initiative poses a number of disadvantages.

First, such a strategy entails high upfront costs. On top of licensing costs for each product in a typical stack (usually half a dozen or more), enterprises often need to purchase new hardware to meet the system requirements of these products. In addition to the upfront costs needed to procure the necessary software and hardware, ongoing maintenance of each of these components requires further capital outlays. A top-down SOA approach thus demands a huge investment of financial resources, with minimal guarantee of a good return.

A second major disadvantage of a top-down approach is that it usually involves a lengthy, multi-year rollout period. A full SOA implementation requires the deployment and configuration of each product in the stack—a complicated process that can take several years to complete. In the interim, development projects and integration needs cannot be properly addressed. Moreover, since a top-down approach demands a monolithic architectural shift, many organizations fall into the “rip and replace” trap,
attempts to simultaneously change existing hardware and software systems as well as development processes in a single shot.

Finally, proprietary SOA stacks are vendor-specific. This means that companies usually have to purchase individual components from, or dictated by, a single vendor, which drastically limits their IT flexibility. Moreover, because proprietary stacks only work with vendor-specific development tools, staffing top-down SOA initiatives becomes a major challenge. The two possible solutions result in increased costs: companies can either hire vendor-specific SOA specialists, who command salaries that are 30% more than general Java developers, or pay for an expensive training program to bring current IT staff up to speed. Even once complete, the scarcity of these developers can create bottlenecks in projects when compared to teams of IT generalists.

Given these pitfalls, it’s not surprising that nearly 80% of all top-down SOA initiatives that were launched over the past decade ultimately failed, resulting in millions of dollars and hundreds of developer hours wasted. For the remaining 20% who successfully implemented their top-down SOA initiatives, the differentiator was the availability of ample financial and human resources to invest in a costly and lengthy infrastructure overhaul. Most organizations, then and now, simply do not have such resources at their disposal. SOA, however, is still critical to business growth and success.

5 “Bottom-Up” SOA

Despite the failures of many top-down initiatives, SOA can be achieved without prohibitive costs or major dislocations to existing IT infrastructures. A bottom-up approach—one that emphasizes incremental adoption rather than wholesale reengineering—is key to success.

In a bottom-up approach to SOA, companies start with a standalone Enterprise Service Bus (ESB) instead of a full proprietary stack. As one of the core components of an SOA stack, ESBs enable the creation and orchestration of services. The main difference is that a standalone ESB can be used without an application server or other infrastructure components, providing a lightweight alternative and eliminating the high upfront costs of implementing SOA with a heavy proprietary stack. And instead of a lengthy rollout period, a standalone ESB can be implemented and deployed immediately to address current needs, enabling developers to build reusable interfaces while also establishing a core framework for integrating with a SOA governance model down the road. Enterprises no longer need to rip and replace their entire infrastructure to achieve SOA, but can adopt it incrementally.

A bottom-up SOA strategy using a standalone ESB provides other advantages as well. Typically, standalone ESBs are built according to open standards, providing organizations with the flexibility to integrate a wide range of systems, applications, and increasingly, cloud services. Many traditional SOA stacks lack such flexibility. Moreover, standalone ESBs do not impose architectural choices or force vendor lock-in, giving companies more IT options when making crucial decisions.
Perhaps most importantly, a standalone ESB does not require specialized developer knowledge or costly training programs. Development teams can start building SOA projects from the bottom-up in a fraction of the time it takes to learn how to use the vendor-specific components and tools of a proprietary stack. This also makes it easier to hire IT staff, control salaries, and onboard new employees in light of the high turnover that is prevalent among development teams. In the end, bottom-up approaches allow organizations to develop an efficient team of generalist developers as opposed to a limited number of “specialists.”

One example of a company that successfully implemented SOA using a bottom-up approach is Nestle. Through its Nespresso business, Nestle pioneered the worldwide market for premium portioned coffee. The company built on its success by incorporating online sales channels. In order to meet dramatic growth forecasts, the company realized it needed a new architecture and integration approach to enable new channels and scale existing ones.

Using Mule at the core, Nestle designed and implemented a new infrastructure architecture according to SOA principles. The new architecture, called Nespresso Open Architecture (NesOA) enables new distribution channels, improves business agility, and scales seamlessly to support high transaction volumes. The company was able to implement the first phase of their SOA initiative in just six months, establishing a flexible platform for quickly and easily making enhancements and extensions in the short- and long-term. To learn more about how Nespresso re-architected their infrastructure to support rapid growth, visit: http://www.mulesoft.com/case-study-nespresso.

A second firm that successfully used a bottom-up approach to SOA is TiVo. A leader in the digital video recorder (DVR) market, the firm had an immediate need to integrate over 40 web services for internal use as well as for integration with partners. The complexity of applications involving these web services both increased the firm’s maintenance costs and limited its ability to onboard new partners.

TiVo selected Mule to build an SOA architecture to address these immediate problems. By adopting Mule, TiVo was able to improve developer productivity by 75%, reduce maintenance costs, and onboard partners faster. At the same time, the firm built a foundation to meet their future needs without taking on a massive project upfront. To learn more about how TiVo gained immediate benefits while setting the stage for future needs, visit: http://www.mulesoft.com/case-study-tivo.

Compared to top-down approaches, organizations adopting bottom-up SOA approaches achieve faster time to ROI, lower risk and better employee utilization. Milestones for bottom-up approaches are measured in products deployed, not consultants hired, developers trained, or components released. In the long run, bottom-up approaches to SOA can produce architectures equal to or greater in benefit than top-down ones. Both can produce strong SOA projects, but by taking an incremental approach instead of a “rip and replace” approach, bottom-up approaches deliver value along the way and allow for more flexible and manageable rollout.
6  Mule ESB: The best way to enable SOA

Mule ESB is the world’s leading integration platform. It is especially well suited for a bottom-up approach to SOA, allowing enterprises to get up and running quickly while laying a strong foundation for incremental adoption of a full SOA infrastructure.

With Mule, enterprises can:

• Expose and integrate existing systems as services to leverage current IT investments
• Address immediate integration challenges while establishing a backbone for SOA initiatives
• Implement SOA at a pace that makes sense for their needs

Unlike heavyweight SOA stacks, Mule is lightweight and flexible, and takes the complexity out of integration. As an open, best-of-breed platform, it can be used standalone or with any other component you choose to build SOA projects. In addition, Mule is easy to learn and understand, allowing any Java developer to become productive quickly without specialized training. It is, simply put, the best way to enable SOA in your organization.

Mule is trusted by many leading organizations that have decided to take a modern, bottom-up approach to SOA. Over 3,200 organizations use Mule in production, including leading companies such as Walmart.com, MasterCard, Nokia, Nestlé, Honeywell and DHL, as well as 5 of the world’s top 10 banks and over 35% of the Global 500. MuleSoft can help you evaluate your key SOA challenges and objectives and develop an agile architecture to help you succeed in coming years.

To learn more and contact an expert, visit: http://www.mulesoft.com/soa-architecture

About MuleSoft

MuleSoft provides the most widely used integration platform for connecting SaaS and enterprise applications in the cloud and on-premise. With the rise of cloud and mobile, enterprises face a choice: become overwhelmed by the resulting explosion of end points or seize the opportunity to gain competitive advantage. Founded on the idea that connecting applications should not be hard, MuleSoft lets organizations harness the power of their applications through integration. MuleSoft’s Anypoint™ technology eliminates costly, time-intensive point-to-point integration, enabling business agility. Delivered as a packaged integration experience, CloudHub™ and Mule ESB™ are built on proven open source technology for the fastest, most reliable integration without vendor lock-in. Supporting billions of transactions per day, MuleSoft is used in production by thousands of enterprises, including Walmart, MasterCard, Nokia, Nestlé and Honeywell, and powers integrations with leading SaaS vendors such as Salesforce.com, NetSuite, Workday, Intuit and Zuora.

For more information: www.mulesoft.com, or email info@mulesoft.com.

Download Mule ESB: http://www.mulesoft.com/download/

MuleSoft and the Mulesoft logo are trademarks of MuleSoft Inc. in the United States and/or other countries. All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

All contents Copyright © 2012, MuleSoft Inc.