



Evolution of SOA

How organizations develop agile SOA to rein in IT costs and secure competitive advantage

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 mega-trends of SaaS, mobile, and Big Data are converging to create a massive explosion of endpoints and data. In this New Enterprise era, businesses can be overwhelmed by the resulting connectivity demands, or they can seize the opportunity and gain competitive advantage by seamlessly connecting everything. The smartest enterprises are transforming this fragmentation into strength by connecting today’s explosion of applications, data, partners and customers into a single, high performing entity.

The strategies for developing this connected enterprise are the fundamentals of agile SOA. The key challenge: finding a platform solution that can deliver immediate impact while providing a foundation for future cloud and on-premise integration demands. Drawing on lessons from leading 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” approaches, and best practices for selecting an SOA solution.

“The smartest enterprises are transforming fragmentation into strength by connecting today’s explosion of applications, data, partners and customers into a single, high performing entity”.

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 code bases resulted in multiple inefficiencies. Code was poorly reused, leading to wasted effort and money spent during development. As infrastructures became more complex, it became

increasingly difficult for developers to maintain and support these applications. Businesses lost agility. If the credit check process needed to be changed, multiple developers updated multiple applications, slowing down the entire modification process. These overly complex and poorly designed applications had substantial impact on 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 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, and customer data services.

“In short, SOA breaks down the islands of business logic and data that are scattered across multiple, disparate applications. In the New Enterprise era, these silos may exist in on-premise or cloud-based software, SaaS applications, or in devices brought from home by employees”.

In short, SOA breaks down the islands of business logic and data that are scattered across multiple, disparate applications. In the New Enterprise era, these silos may exist in on-premise or cloud-based software, SaaS applications, or in devices brought from home by employees. SOA enables interoperability across all silos 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.

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:

Business objectives

- 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?
- What new business services do I plan to offer (e.g. SaaS or APIs)?
- Does my current infrastructure support agile growth and provide enough flexibility to quickly pivot to changing business needs?
- Which SOA solution provides the best return on investment (ROI)?

Technical considerations

- Does my SOA solution provide support for all of the technologies in my IT environment?
- How does my SOA solution integrate on-premise and cloud applications? Can both types of endpoints be integrated on a unified platform?
- Can my SOA solution be used with application servers, development tools, etc. from different vendors or will it force vendor lock-in?

Project requirements

- What is the timeline of my SOA initiative? How quickly can I implement my solution and be productive?
- What are the upfront licensing costs of my SOA solution? What is the total cost of ownership (TCO), including additional hardware and software costs ongoing maintenance, developer training, etc.?
- How will I staff my SOA project? Do my in-house developers possess the knowledge and skills required to implement my solution or will they need additional training?
- Does my project require the expertise of an external consultant? How much will that cost?

“Top-Down” SOA

When enterprises first adopted SOA, many opted for a “top-down” approach. This entailed launching a single organization-wide SOA initiative and selecting a 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. The company’s development team would then 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 skill sets 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.

“ Since a top-down approach demands a monolithic architectural shift, many organizations fall into the “rip and replace” trap, attempting to simultaneously change existing hardware and software systems as well as development processes in a single shot”.

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, attempting 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 a single vendor, which drastically limits IT flexibility. As many proprietary stacks lack cloud extensibility, this may prevent New Enterprise organizations from integrating mission critical endpoints. Moreover, because proprietary stacks only work with vendor-specific development tools, staffing top-down SOA initiatives becomes a major challenge. 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 re-engineering—is key to success.

“ 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”.

In a bottom-up approach to SOA, companies start with a standalone Enterprise Service Bus (ESB) or integration platform as a service (iPaaS) instead of a full proprietary stack. As one of the core components of an SOA stack, ESBs and iPaaS solutions enable the creation and orchestration of services without requiring an application server or other infrastructure components. This provides a lightweight alternative and eliminates the high upfront costs of implementing SOA with a heavy proprietary stack. And instead of a lengthy rollout period, a standalone ESB or iPaaS 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 or iPaaS 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.” developers as opposed to a limited number of “specialists.”



One example of a company that successfully implemented SOA using a bottom-up approach is Nestlé. Through its Nespresso business, Nestlé 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 ESB at the core, Nestlé 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 leading UK food retailer and FTSE 500 company also used a bottom-up approach to SOA with a cloud-based integration solution. Operating in a competitive retail market, the company

needed innovative ways to drive in-store revenue while also expanding into new product categories and channels on the web. As critical data exists in both cloud and on-premise applications, the SOA solution they identified also needed extensibility on both sides of the corporate firewall.

To bring existing customer and product data into the company's new Salesforce solution, a popular SaaS CRM application, the company selected CloudHub, the enterprise-class integration platform as a service (iPaaS) from MuleSoft. CloudHub delivers real-time application integration of their on-premise and SaaS applications. As a result, the company has a single view of the customer to support multi-channel marketing online, in the call center, and at the supermarket checkout register. To learn more about how this UK retailer gained immediate benefits while setting the stage for future needs, visit: <http://www.mulesoft.com/case-study/sso-integration>.

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 & CloudHub : The best way to enable SOA

MuleSoft provides the world's leading integration platform. Mule ESB and CloudHub are 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 ESB & CloudHub, 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
- Connect on-premise, cloud, and SaaS applications on a unified integration platform
- Develop once, deploy anywhere with consistent, easy to use development tooling

Unlike heavyweight SOA stacks, Mule ESB 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. When combined with CloudHub and the other components of the Anypoint(TM) platform, MuleSoft offers the only unified integration platform for connecting anything, anywhere. It is, simply put, the best way to enable SOA in your organization.

MuleSoft is trusted by many leading organizations that have decided to take a modern, bottom-up approach to SOA. Over 3,500 organizations use Mule ESB 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>

Further resources

Blog posts and resources

SOA school

<http://blogs.mulesoft.org/soa-school-service-orchestration-2>

SOA School: Governance 2.0 with Anypoint Service Registry

<http://blogs.mulesoft.org/soa-school-governance-2-0-with-anypoint-service-registry/>

Why open source SOA makes sense

<http://www.mulesoft.com/resources/esb/open-source-esb-best-choice-soa/>

Webinars

The mega-trends of SaaS, mobile and Big Data are converging, generating a new wave of business opportunity for enterprises. The convergence demands a new kind of platform – one that connects and takes advantage of the explosion of endpoints and data caused by organizations each choosing a uniquely diverse set of best-of-breed applications to power their business. Ross Mason will discuss his vision for this new platform and demonstrate how MuleSoft's solutions are making it a reality.

<http://www.mulesoft.com/webinars/connecting-the-new-enterprise>

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.



www.mulesoft.com



info@mulesoft.com



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 © 2013, MuleSoft Inc.