



MuleSoft®



Architecting the Composable Enterprise *IT for the 21st century*

Jason Bloomberg, President, Intellyx LLC

The Demise of Centralized IT

Enterprise IT was much simpler in the days before the cloud. Big companies owned their own technology and purchased enterprise licenses for many of the applications anybody in the organization might care to use. To connect everything together, IT executives would buy middleware – which naturally went in the middle of everything.

Furthermore, as Conway's Law would suggest, the organizational structures of the IT shop followed the centralized lines of the technology architecture. Simple hierarchical org charts would suffice, as different groups dealt with different tasks.

Outside this centralized oasis of technology? The vast unknown territory known to techies as *the business* – as though enterprises clove neatly into two parts, the tech cognoscenti here and everybody else over there.

Then along came the cloud. And social media. And mobile technologies. The world of IT has never been the same since. Not only are technologies distributed well beyond the perimeter of the enterprise as cloud computing has given birth to vast ecosystems of third-party technology providers, but the organizational principles have likewise transformed.

Technology has infiltrated every corner of the business – if we can even differentiate *the business* from IT anymore. Today's enterprises are software-driven organizations, shifting the roles and perspectives of everyone in the organization, whether they be "business" or "technology."

The opposite of centralized IT, therefore, isn't only decentralized IT – although distributing the technology effort in the enterprise is an important aspect of this trend. We're also decentralizing how the business accesses and uses technology resources as well.

The decentralized IT organization must build out the capabilities the business needs, while empowering business users to get their jobs done and thereby meet the needs of customers and the organization.

The end state of this combination of decentralization and empowerment is the *composable enterprise*: where people can support the changing demands of the organization by assembling and reassembling modular components – components made up of both technology and people.

Supporting this notion of the composable enterprise introduces profound business challenges for both business and technology leadership. If they don't take advantage of such cutting-edge approaches to addressing customer demands, they will fall behind or go out of business.

The organizational challenge at the heart of the composable enterprise centers on realigning every line of business and department along customer-driven lines. The technology challenge: how do you take crusty, monolithic legacy technology and unbundle it into components in order to assemble them in different ways, while retaining its value?

The Challenges of First Generation SOA

This notion of IT delivering composable capabilities to the business, as though application functionality and data were LEGO blocks for business people to assemble, is not a new idea. In fact, composability was an important benefit of Service-Oriented Architecture (SOA) – at least, the version of SOA IT shops struggled with during the 2000s.

SOA is an approach for abstracting enterprise software capabilities as reusable services in order to support more flexible business processes and ideally, more agile organizations. However, in retrospect, the original promise of SOA was largely unrealized.

Vendors used the approach to sell middleware, which led to expensive and difficult implementations. Deployments were centralized, leveraging hub-and-spoke technology that was ill-suited for the cloud. The SOA organization was centralized as well, with activities taking place internal to the organization, limiting the ability for teams at different organizations to share knowledge of lessons learned.

"Technology has infiltrated every corner of the business – if we can even differentiate the business from IT anymore. Today's enterprises are software-driven organizations, shifting the roles and perspectives of everyone in the organization,

Eventually, the architectural focus on improving IT and organizational governance in order to achieve greater levels of business agility was largely subsumed into the technical minutiae of enterprise integration. The entire SOA exercise, therefore, became an exercise in tight coupling, connecting endpoints to endpoints.

The principles of SOA were solid, but most implementations were overengineered and too complicated. Web Services proved to be too challenging. And worst of all, enterprises failed to emphasize the *consumers* of the services – both in the sense of the software endpoints that interacted with services, as well as the people who would want to consume service capabilities and compose them to solve business problems.

The Rise of API-led Connectivity

The first generation of SOA centered on the role of the Enterprise Service Bus (ESB), and sported Web Services as the primary type of service. However, among the main roadblocks preventing greater success with SOA were the complexity and technical limitations of Web Services. Even though these XML-based standards promised loose coupling, most Web Services deployments were nevertheless excessively tightly coupled, limiting flexibility overall.

The rise of cloud computing also shifted the SOA discussion. What cloud computing brought to the SOA table were the principles of horizontal scalability and elasticity, automated recovery from failure, eventually consistent data (or more precisely, tunable data consistency), and a handful of other now-familiar architectural principles. The cloud also emphasized the importance of decentralized computing.

The appearance of the cloud coincided with the exploding popularity of Representational State Transfer (REST). REST arose largely out of

the ashes of Web Services, and helped organizations overcome many of the SOA roadblocks that had limited their success with the architecture.

As a result, second-generation SOA was REST-based and cloud-friendly, favoring lighter weight approaches to moving messages around than the heavyweight ESBs that gave SOA a bad rep. RESTful interfaces cleaned up a lot of the mess that Web Services left behind, as they were web-centric, lightweight, and far easier to use than Web Services.

The JavaScript Object Notation (JSON) also played an important role in the maturation of SOA, as it proved to be a simpler and more flexible data format than XML. The fact that JSON objects were themselves JavaScript also provided an ease of use that XML was ill-suited to deliver.

The combination of REST and JSON essentially moved the ball on application programming interfaces (APIs), leaving behind the challenges of Web Services. Today, APIs are more likely to be RESTful, HTTP-based interfaces than SOAP-based Web Services.

In fact, perhaps the most successful part of REST to date has been the simplification of the API. Enterprises no longer need a language-specific protocol that depends upon sophisticated network controls under the covers. Today they can take HTTP for granted, and a simple request to a URL suffices to establish any interaction they care to implement between any two pieces of software, regardless of language.

As a result, today's APIs are inherently Web-centric. They take advantage of Web-based, cloud-centric protocols and architectural principles, like horizontal scalability and automated recovery from failure. And most importantly, RESTful APIs' Web-centricity shifts the focus of enterprise integration to the consumer.

The Web, after all, is more about the human being interacting with a Web endpoint than it is about the Web servers, or any of the other infrastructure behind the scenes. It's no wonder, therefore, that today's digital transformations depend upon this Web-centricity to satisfy the needs and preferences of the customer.

APIs and the Composable Enterprise

The customer-centricity of digital efforts has led to the broader trend of the democratization of technology. No longer can enterprise apps

afford to be immune to the pressures of consumer demands. Instead, everyone in the enterprise – or any other size organization, for that matter – expects the applications they use at work to be as convenient, flexible, and mobile-enabled as the apps they use anywhere else.

Enterprise IT, therefore, must respond to this rising tide of democratization, not only by supporting mobile interfaces to enterprise apps, but also by empowering an entire ecosystem of digital capabilities for both internal and external users. Such users expect to find apps in app stores – marketplaces of functionality, originating both inside and outside the organization.



(Photo credit: [Joe Miserendino](#))

Furthermore, users are expecting to compose the functionality of APIs and the data they provide to meet shifting business needs. Sales data should connect to business intelligence should connect to analysis and visualization – the list of such composition opportunities is endless. This expectation of composability is at the heart of the composable enterprise.

From a technical perspective, the secret sauce that makes this app store-driven, user-centric composition vision come to life are the APIs that form the glue among the various application components and other services, including the small, modular components called microservices. And now that such APIs are web-centric, leveraging REST, JSON, and other easy-to-use protocols, there's no excuse for developers not to get them right.

Furthermore, there are so many available APIs that meet the needs

and business models of so many organizations that an *API economy* has grown up around them. In this API economy, developers and other people can assemble apps from a mix of components built in-house and available in the cloud.

Companies who may have never thought of themselves as offering software-based products or services to their customers are now able to leverage APIs to expand their offerings. As enterprises in multiple industries become software-driven organizations, APIs become the means for providing value to customers, for maintaining efficient relationships with suppliers, and for participating in the broader commerce communities to which they belong.

New Zealand Post is a great example of such an organization. This company developed a number APIs, empowering external users to integrate with its shipping, addressing, and postal systems. These APIs helped to build its parcel delivery business, which overtook mail delivery as the largest source of revenue in 2014. New Zealand Post is now using a commercial version of its addressing API to assist with identity verification on credit card applications.

Architectural Leadership for the Composable Enterprise

APIs support enterprise composability at two levels. Inside the organization, people leverage APIs at the team level, while external to the organization, customers and partners can mix and match APIs from different places, composing and recomposing capabilities and information at will.

In some cases, the business drives composability top-down. Executive-level concerns drive strategic initiatives that drive API-led composability. In other cases, composability is bottom-up, as developers put together capabilities to meet various project needs. However, in either case, the same APIs facilitate such composability – assuming, of course, they are architected properly to support such composability.

Architecture, in fact, is a critical enabler of the composable enterprise vision. APIs must be reusable and modular, and it goes without saying that nonfunctional requirements like compliance and security must be bulletproof. All of these requirements depend upon a lightweight, Web-centric architecture.

Where, then, should the IT organization go to get proper architectural leadership – or any other expertise necessary to execute on the composable enterprise? Perhaps a center of excellence will do.

Centers of excellence (CoEs) are teams and associated knowledge resources that provide leadership, best practices, research, and support for a focus area like architecture or APIs. On the surface, it sounds like such a center is just the ticket to help an organization transition to becoming a composable enterprise.

There's just one problem with this plan: CoEs are by definition *centralized*. They actually become roadblocks for IT, as well as for line of business people who want access to IT capabilities and data, because leveraging a center of excellence requires formal requests to a small, typically overworked team.

In contrast, the decentralized alternative to a COE is a *center for enablement*. Centers for enablement focus on rolling out new capabilities and assets to a

skilled audience – often with the support of a self-service capability like a portal or app store.

Instead of acting as an ivory tower repository of expertise, a center for enablement distributes templates that various audiences can use as starting points. The goal is to teach people 'how to fish' as well as '*where* to fish' for the expertise and capabilities they require.

Templates, APIs, app stores, and marketplaces all facilitate self-service access to IT resources. However, moving to a self-service model requires changing behavior in the organization. Get this model right, however, and it frees IT to focus on more important tasks.

IT for the 21st Century: Supporting the Composable Enterprise

The vision of the composable enterprise is a vision of a world where IT is a partner to the business. IT is no longer a back office function, but rather an enabler of the business. To make this vision a reality, the approach can't only be top-down or bottom-up. It must be

customer-driven and end-to-end – the essence of digital transformation.

At one end, of course, are legacy systems, which will continue to present a challenge to the composable enterprise. However, rip and replace is rarely the best option. Instead, expose legacy assets as APIs, and modernize them how and when delivers the most value to the organization and its customers.

APIs make even inflexible legacy assets composable as part of the API economy. APIs are the lynchpin of a modern approach to integration that will create flexibility, agility, efficiency, and ultimately business success for years to come.

The business no longer has the time to wait for centralized IT. But that doesn't mean the IT organization goes away. Instead, there are the three roles that remain critically important for modern IT: security, governance, and maintaining access to systems of record – via APIs.

The role of architecture is shifting as well. The top priority for architecture is supporting the organization's business agility goals – helping the organization deal with the change at the heart of the composable enterprise.

In order to successfully become composable enterprises, organizations must decentralize the IT organization and support centers of enablement rather than centers of excellence. Implement lightweight, web-centric architectures and the design principles of API-led connectivity.

Such business and technological transformation is difficult, but the path to success is clear. Every organization has the power to become a composable enterprise.

Copyright © Intellyx LLC. Intellyx advises companies on their digital transformation initiatives and helps vendors communicate their agility stories. MuleSoft is an Intellyx client. Intellyx retains full editorial control over the content of this article.



MuleSoft's mission is to connect the world's applications, data and devices. MuleSoft makes connecting anything easy with Anypoint Platform™, the only complete integration platform for SaaS, SOA and APIs. Thousands of organizations in 60 countries, from emerging brands to Global 500 enterprises, use MuleSoft to innovate faster and gain competitive advantage.