

What You Need to Know About Service-Oriented Architecture

Think of a Service-Oriented Architecture (SOA) as a system of gears: some big and slow-turning, some small and fast. SOA is the right mechanism—a transmission of sorts—for an IT environment (like so many others) in which relatively ponderous data-crunching legacy systems must mesh with agile front-facing applications.

Lately, SOA has been gaining traction, especially as Web services have been maturing. Gartner estimates that by 2008, more than 60 percent of enterprises will use SOA as a "guiding principle" when creating mission-critical applications and processes.

What the Heck Is an SOA?

SOA's start with services, which are groups of software components that carry out business processes, for example, verifying a credit card transaction or processing a purchase order. At its most basic, an SOA is a collection of services on a network that communicate with one another. The services are loosely coupled (meaning that an application doesn't have to know the technical details of another application in order to talk to it), have well-defined, platform-independent interfaces, and are reusable. SOA is a higher level of application development (also referred to as coarse granularity) that, by focusing on business processes and using standard interfaces, helps mask the underlying technical complexity of the IT environment.

What Are the Benefits of Adopting an SOA?

- SOAs make it easier to integrate IT environments.
- SOA works very well in heterogeneous environments. Developers don't have to spend an inordinate amount of time writing new lines of code to connect applications. Instead, they can use standard protocols, such as Web services. And large chunks of SOA code are reusable, reducing development costs.
- You don't need to rip and replace legacy systems with brand-new ones. By identifying the capabilities of existing systems and leveraging them, you maximize the value of your IT investments while minimizing your risk.
- Also, building services—for example, using simple object access protocol (SOAP) and Web services description language (WSDL)—not only smooths the internal integration process, it also lets customers and business partners share information more easily across company firewalls.

Another benefit of an SOA is that it can lead to a better dialogue between the CIO and line-of-business execs by forcing IT workers to think in terms of business—not technical—architectures. If a business wants to build a better inventory control system, for example, the operations folks can hook up with the IT architects and talk about the best way to design it based on business flows and how best to meet the needs of the business.

For that dialogue to work, businesspeople have to think about the best ways to run their business. What processes do I need to put in place to best accommodate my customers? How can I improve my level of customer service?

What Role Does Web Services Play in an SOA?

First, it's important to note that an SOA does not require Web services; and Web services can be deployed without an SOA. There are those, however, who believe that building an SOA using Web services is the ideal approach.

What Are the Challenges?

Security is a big one. You must deal with the lack of security standards for Web services. To overcome some of these security roadblocks, you should move slowly when setting up an SOA, focusing first on business processes that don't require a high level of security.

Another issue is network monitoring. With orchestrated, complex, Net-centric business processes in a service-oriented architecture, we have complex monitoring and auditing requirements. For instance, when a transaction goes awry on a service-oriented network, which could involve multiple service providers, finding out what went wrong or where the transaction dropped or whether someone put bad information in the network can be a challenge.

Are There Any Generally Accepted Best Practices for Building an SOA?

It may sound obvious, but having a blueprint for your SOA is critical. It's very easy for companies, especially large enterprises with disparate operations, to buy new technologies or integrate applications without regard to how they fit into the overall plan. The challenge in building an SOA is to keep people—including both IT and business-side staff—focused on the architecture goals.

IT execs will also need to identify the right level of service to provide. And those services shouldn't have too fine a granularity—that defeats the goal of services, which is to function at a higher, business-process level. Too narrow a focus creates a need for more services, which increases development time. And in the worst case, too many services can flood a network. You should also employ an SOA where it will do the most good.

Based on an article by Todd Datz in Jan 15 2004 issue of CIO Magazine.