

Lesson 5

SOA



■ Reading list

- Service Oriented Architecture FOR dummies, Judith Hurwitz, Robin Bloor, Marcia Kaufman, and Dr. Fern Halper, 2ND EDITION

What is a Service Oriented Architecture (SOA)?

- A method of **design, deployment**, and management of both **applications** and the **software infrastructure** where:
 - All software is organized into business services that are network accessible and executable.
 - services are provided to the other components by application components.
 - A service is a discrete unit of functionality that can be accessed remotely and acted upon and updated independently.
 - E.g. retrieving a credit card statement online.

SOA is a business approach

- to building IT systems that allows businesses to
 - Leverage existing assets
 - Create new ones
 - Easily enable the inevitable changes required to support the business W

Smarter Business and Smarter IT

- **business** and **IT** have a means to meet each other halfway and work together using a business focused approach to develop new ways to use technology to grow the firm.
- SOA can make it **easier** and faster to build and deploy IT systems that directly serve the **goals** of a business
- SOA helps companies to **develop tools** they need to spot new trends and opportunities and see new ideas to fulfill their needs.

Key Characteristics of SOA

- Quality of service, security and performance are specified.
- Software infrastructure is responsible for managing.
- Services are cataloged and discoverable.
- Data are cataloged and discoverable.
- Protocols use only industry standards

Clarification

- 1. SOA is for building business applications**
 1. Not all applications, just business oriented
- 2. SOA is a black-box component architecture**
 1. Deliberately hides complexity wherever possible
 2. Enables the reuse of existing business applications by adding a fairly simple adapter to them
- 3. SOA components are loosely coupled**
- 4. SOA components are orchestrated to link together through business processes to deliver a well-defined level of service.**

Common problem to many large companies

- have lots of similar programs
 - Every department has its own version of the software
- Across a particular company, you might find lots of different versions of more or less the same process
- Such duplication becomes a nightmare when one company acquires another and finds that they have **similar** (but not identical) applications purporting to do the same thing.
- These small variations create very complicated and expensive to maintain
 - Small changes can create big nightmares
 - It's difficult to find every instance in every application that needs to be changed
- This type of change takes time away from getting to market quickly with new products.

Solution : reuse

- With SOA, these important business processes become **business services**
 - Creating an invoice
 - Calculating an interest rate
 - Securing a reservation
- one single business service for a given function that gets used **everywhere** in your organization.
- With SOA, when you need to change a business policy, you **change** it in only **one place**. And, because the same service is used everywhere, you have consistency throughout your organization.

What is a “Service”?

- A Service is a reusable component.
 - A Service changes business data from one state to another.
 - A Service is the only way how data is accessed.
 - If you can describe a component in WSDL, it is a service
 - The Web Services Description Language is an XML-based interface definition language that is used for describing the functionality offered by a web service.
- Service

Assuring a Better Quality of Service

- smooth, satisfying, efficient operation is what businesses can achieve by using service oriented architecture.
- SOA adds predictability and regularity between business rules, policy, and software services. Therefore, one of the greatest selling points for SOA is that it can help management know what tasks a particular service is executing and what rules and policies are codified within these services.
- Being able to track this not only makes software within the company better but also makes corporate governance more predictable and less cumbersome.
- When change happens, you can be a lot more agile with SOA.

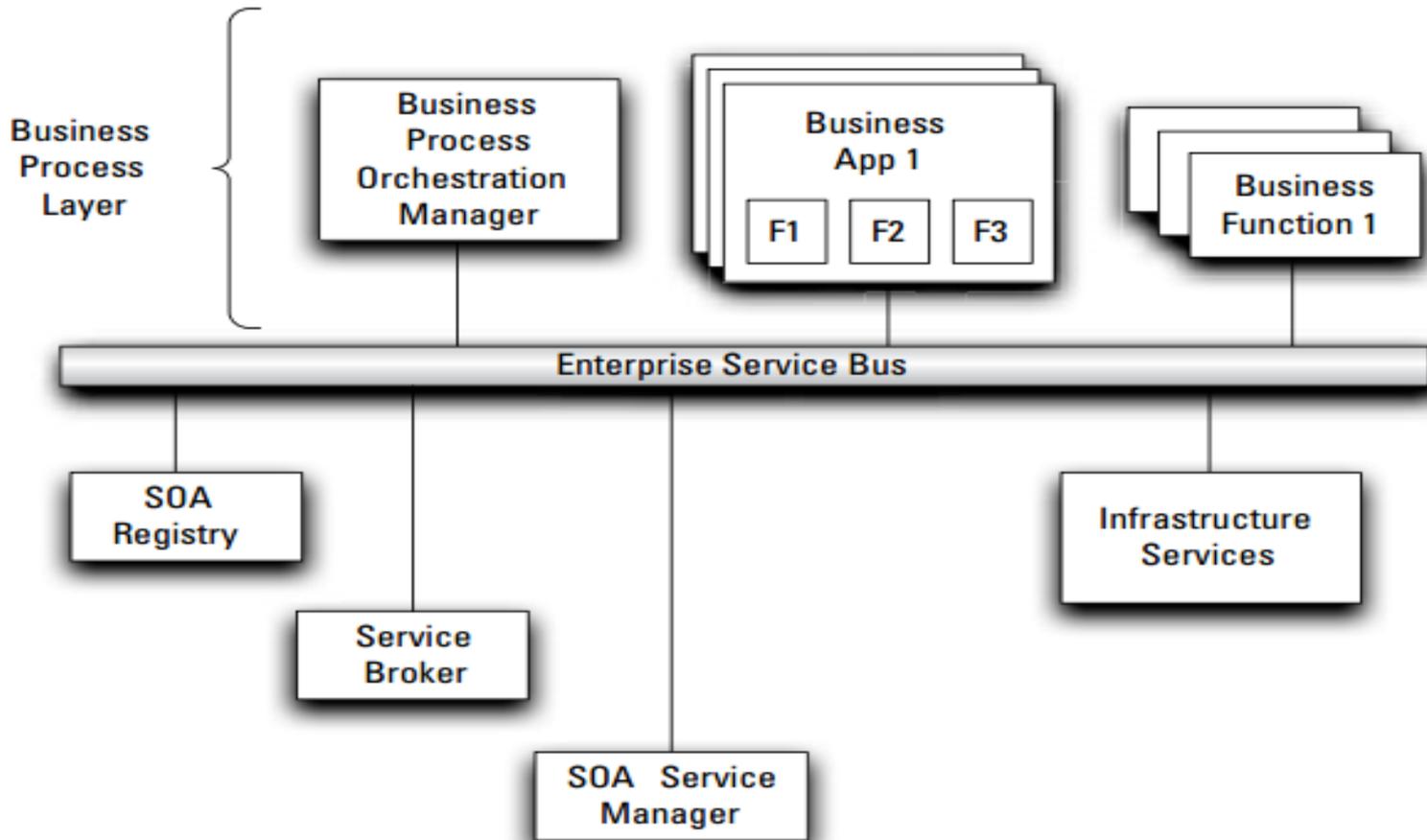
When designing a SOA

- They must be **safe**. Safe means that the service itself is secure and doesn't introduce bugs and problems into the organization.
- They must be **accurate**. Accuracy means that the service itself executes the function it's designed to execute. Organizations implementing SOA must be reassured that each business service is executing the **right function**
- They must be **predictable**. Predictable means that the service does what it's expected to do. If a service is designed to calculate a 30-year mortgage, it had better do exactly that each time it's used

Layers of SOA

- Business Services layer
 - Contains your business logic
 - What software applications provide
- Plumbing layer
 - deals with your computing resources.

Fundamental SOA components



The Enterprise Service Bus (ESB)

- The ESB makes sure that messages get passed back and forth between the components of a SOA implementation.
- To transport the messages between software components
- the ESB is represented as a separate pipe through which information and instructions flow.
 - In reality, it's not really a pipe. Rather, the ESB is a collection of software components that manage messaging from one software component to another.

Registry

- Think of the SOA registry as a kind of electronic catalog where you store information describing what each component does. It has two roles:
- Rooted in the operational environment
 - Provides reference information about software components that are running or available for use.
 - The software in a SOA framework that brings components together by using the rules associated with each component.
- Rooted in the world of programmers and business analysts
 - Acts as a reference that helps them select components and then connect them to create composite applications that represent business processes.
 - Stores information about how each component connects to other components. In other words, the SOA registry documents the rules and descriptions associated with every given component

Repository

- Feeds the service oriented architecture with changes and new components.
- Works within the operational environment and takes on the responsible role of acting as the counterpart of the registry within the development environment.

Difference between repository and registry

- **Repository:** Central reference point for all the components within the software development environment from which services are built
- **Registry:** Central reference point for definitions, rules, and descriptions associated with every service within a SOA environment

Business process orchestration manager (BPOM) and service broker

- BPOM provides the technology to connect people to people, people to processes, and processes to processes;
- The service broker connects services to services, which in the end, enables the flow of business process.

The role of the SOA service manager

- is to make sure that the technology underneath the SOA environment works in a consistent and predictable way.
- The goal is to create an environment where all these components work together to improve the flow of business process.
- All these services are required to link unrelated technology components as though they were designed to work together.

SOA vs Cloud Computing

- *Service Oriented Architecture (SOA)*
 - *is a flexible set of design principles used during the phases of systems development and integration. A deployed SOA-based architecture will provide a loosely-integrated suite of services that can be used within multiple business domains.*
- *Cloud Computing*
 - *is Internet-based computing, whereby shared resources, software and information are provided to computers and other devices on-demand, like a public utility.*

similarities

- First, both emphasize the service concept.
 - By definition, a service is performance of work by one for another. Both CC and SOA delegate work to other parts of the system, either by the service provider or other business components. With that delegation, people can use the services without worrying about the implementation details and scalability. Most important, services in CAA and SOA can be shared by multiple applications and users, thus optimizing resource utilization.
-
- Second, both promote loose coupling.
 - Each architecture demands minimum dependencies among different parts of the system. As a result, any single change on one part of the system has limited impact on the overall system.

Differences

- **Horizontal versus vertical services.**
 - The services in SOA mainly focus on business. Each service may represent one aspect of the business. Combined together, these services consist of a business application or solution. In this sense, the services are horizontal.
 - The services in CAA are mainly layered according to typical software stacks. The lower services support the upper services to deliver applications. Therefore, I call them vertical services.
- **Application vs. infrastructure**
 - SOA is for application architecture. The dividing of different components is based on their roles in the SOA applications. More often than not, you start with a business problem and then abstract out the services. These services can be re-used by other applications in the future.
 - Cloud Computing is for IT delivery. The dividing of different services is based on their roles in a software stack that is mostly well defined. You don't need a problem before defining the cloud services. The services can be easily re-used by all applications.

conclusion

- SOA and cloud computing share many common principles, but also differ significantly in their role in IT architecture.
- SOA is mainly an application architecture with horizontal services; while cloud computing is an IT architecture with vertical services.
- Given the differences, SOA and cloud computing complement each other very well

References

- **Service Oriented Architecture FOR dummies**, Judith Hurwitz, Robin Bloor, Marcia Kaufman, and Dr. Fern Halper, 2ND EDITION
- **What is a Service Oriented Architecture?** Prof. Paul A. Strassmann, George Mason University, November 19, 2007
- **Service Oriented Architecture (SOA)**, *David Linthicum*, <https://msdn.microsoft.com/en-us/library/bb833022.aspx>
- **SOA and Cloud Computing: Are They The Same?** , Steve Jin, 2010, <http://blogs.vmware.com/vcloud/2010/04/soa-and-cloud-computing-are-they-the-same.html>