Service **O**riented **A**rchitecture

&

Enterprise **S**ervice **B**us

Authored by Chris Bohnet

Agenda

What is Service Oriented Architecture (SOA)?

▸ A service is a self-contained part of functionality.

▸ Each service uses as little complexity as possible.

▸ Many services are combined to form applications on a large scale.

▸ The services are provided to each other through a communication protocol (HTTP) over a network.

▸ A service provider makes a service available to use by means of a registry and contract that describes the service.

▸ A service consumer locates services in the registry and builds the client component in order to use the service.

What is Enterprise Service Bus (ESB)?

▸ It's the communication between components of a SOA.

▸ ESB facilitates the transfer of data and instructions amongst services, data warehouses, internal systems, processes and more.

▸ The ESB integrates the many different applications into a single bus-like infrastructure.

▸ It can be thought of as the bridge that brings components together to communicate.

▸ The applications talk to the bus and the bus communicates the data back and forth between the user and the provider.

▸ It's the message broker which does protocol conversion, message format transformation and routing.
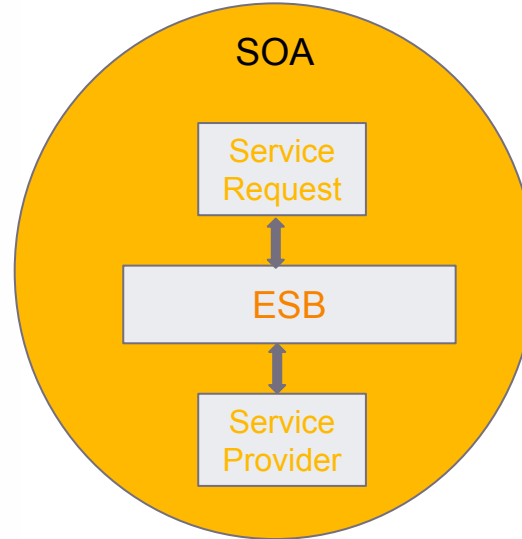
Relationship between ESB and SOA Architecture

ESB

Technical Approach
ESB is an implementation that works off the principles of SOA and supports the SOA approach.

Communication Backbone
The ESB allows the SOA to be decoupled services to exchange and share data amongst each other.

SOA

Service Request

ESB

Service Provider

SOA

Architectural Approach
SOA is an architectural approach to expose services in a granular manner.

Decoupled Services
Services interact with each other regardless of the service type, so they can be platform or protocol specific but can still interact and exchange data.

Data transmission through SOA architecture
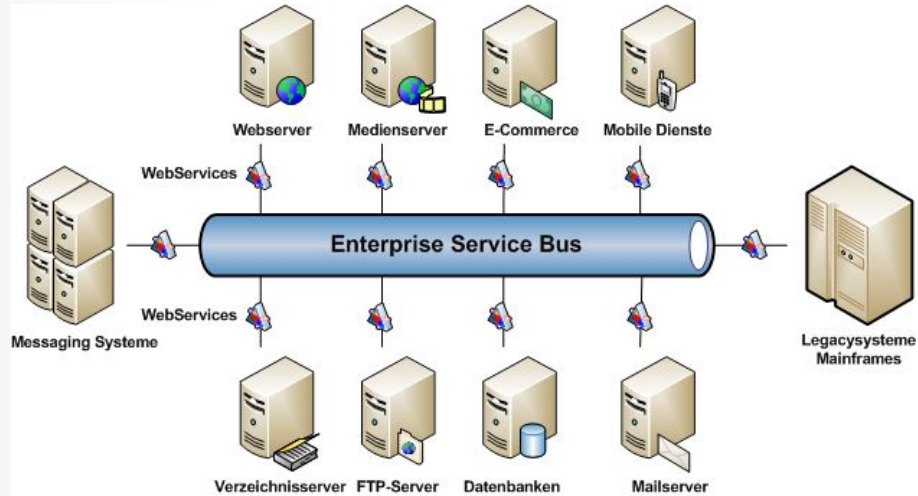
Data is requested from any number of types of services and sent through the ESB.

The service bus has various adapters for the services to provide a common communication platform for data exchange.

Data is evaluated from the provider and response returned to the consumer via the service bus.

Consumers

Service Bus

Providers

Advantages

Reusing services results in decreased development time and cost.

Scalable in as consumers increase additional servers can be added.

Can integrate different services with different protocols and platforms.

SOA

SOA serves for parallel development easing work dependencies.

Services are smaller so easier to debug or test independently of other services.

Modular design provides ease of maintenance and flexibility and scalability.

Disadvantages

Increased overload with all inputs being validated before sent to service.

High bandwidth server needed for web services with millions of daily requests.

SOA that use XML require additional program analysis and composition.

SOA

Not suitable for GUI functionalities since SOA is heavy on data exchange.

Some services require large amounts of executables to perform tasks.

Doesn't involve synchronized thinking which may result in illogical parts in the model.
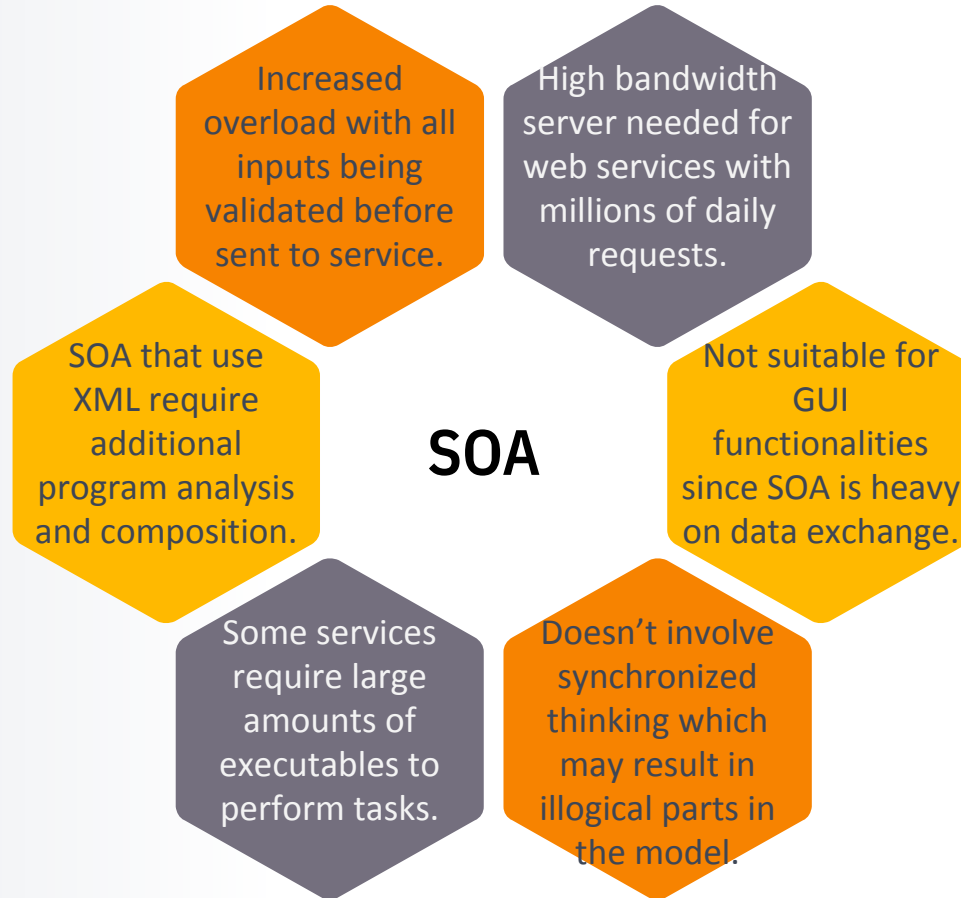
SOA

Deployment

Service providers register their implementation to a registry that is discoverable by service consumers.

The services function based on a formal definition or contract (WSDL) that is independent of the platform and language.

The definition hides the implementation of the service.

The deployment process allows SOA systems to be deployed independently of development.

Implementations can use one or more protocols and/or a wide range of technologies, such as web services based on WSDL & SOAP, Messaging, HTTP and others.
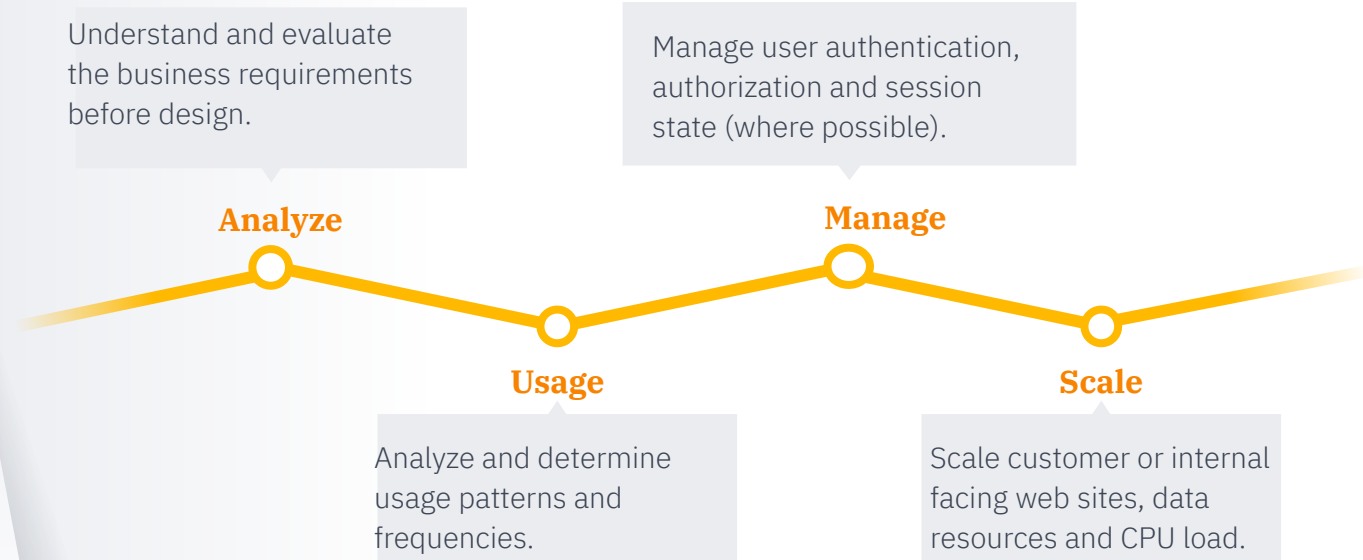
Maintenance

Services are independent of each other so then can be updated and modified without impacting other services.

Code is easier to maintain than traditional architectures since the services are built on smaller modules.

Easier to maintain since services are available for any platform and are easily discoverable.

How to scale a SOA architecture

Break down the complex tasks to smaller processes that operate independently.

Understand and evaluate the business requirements before design.

Manage user authentication, authorization and session state (where possible).

Analyze

Manage

Usage

Scale

Analyze and determine usage patterns and frequencies.

Scale customer or internal facing web sites, data resources and CPU load.

Resources

Geeks for Geeks, "Service Oriented Architecture", as referenced at
https://www.geeksforgeeks.org/service-oriented-architecture/.

TechTarget, "Enterprise Service Bus (ESB)", ad referenced at

https://searchapparchitecture.techtarget.com/definition/Enterprise-Service-Bus-ESB

DifferenceBetween.com, "Difference Between SOA and ESB", as referenced at
https://www.differencebetween.com/difference-between-soa-and-vs-esb/.

Dr. Dobbs, "Scaling SOA with Distributed Computing", as referenced at
https://www.drdobbs.com/web-development/scaling-soa-with-distributed-computing/193104809.

JavaWorld, "What is service oriented architecture?", as referenced at
https://www.javaworld.com/article/2071889/what-is-service-oriented-architecture.html.

Sigma, "Deployment of SOA infrastructure", as referenced at
https://sigma.software/about/media/deployment-soa-infrastructure.

IT Release, "Advantages and Disadvantages of service oriented architecture (SOA)", as refereneced at
http://www.itrelease.com/2018/10/advantages-and-disadvantages-of-service-oriented-architecture-soa/.

The art of service, "The disadvantages of SOA", as referenced at
https://theartofservice.com/the-disadvantages-of-soa.html.

Mule ESB, "What is ESB?", Diagram used as referenced at
http://muleesb.blogspot.com/2012/04/what-is-esb.html.