Enterprise Systems and Services

Service Oriented Architecture

What defines SOA?

- Is a logical representation of a repeatable business activity that has a specified outcome.
- Is self-contained.
- May be composed of other services.
- Is a "black box" to consumers of the service.

Generally large SOA systems are built in layers.

Components of SOA

Consumer Layer

- These are GUI for end users or apps accessing apps/service interfaces.
- Not always included.
- Generally written in Javascript for Modern Web applications but can be in JSP, PHP, .NET.

Business Process Layer

- These are choreographed services representing business use-cases in terms of applications.
- Track a package, Customize a shipment, etc.

Services

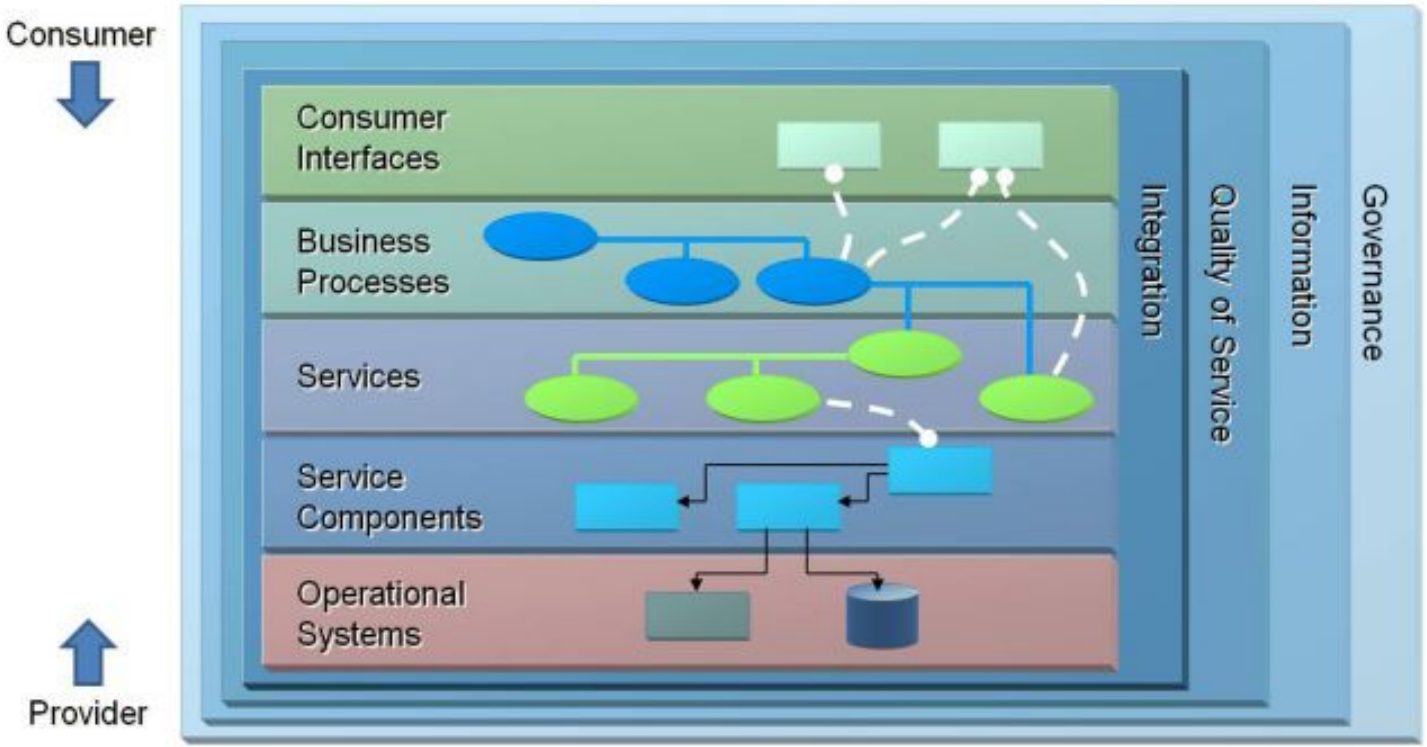
- Services are consolidated together for whole-enterprise in-service inventory.
- Pickup Service, Tracking Service, Rates Service, etc.
- Services can call each other but this increases coupling.

Service Components

- These are simply the components that make up a service. These may be shared as well.
- Address Book, Customer Information, technical libraries, common frameworks.

Operation Systems

- This layer contains the data models, enterprise data repository, technological platforms etc.
- Databases, Security and authorization systems, monitoring, and everything else needed to keep higher layers up and running.


Consumer
Consumer Interfaces
Business Processes
Services
Service Components
Operational Systems
Integration
Quality of Service
Information
Governance
Provider

SOA Principles

Standardized Service Contract

Services adhere to a communications agreement, which is usually described via a document or even another service.

Loose Coupling

Services maintain a relationship that minimizes dependencies and only requires they maintain an awareness of each other.

Abstraction

Beyond descriptions in the service contract, services hide logic from the outside world.

Reusability

Logic is divided into services with the intention of promoting reuse.

Autonomy

Service have control over logic they encapsulate, from a Design-time and a Run-time perspective.

Statelessness

Services minimize resource consumption by deferring the management of state information when necessary.

Discoverability

Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted.

Composability

Services are effective composition participants, regardless of the size and complexity of the composition.

Granularity

A design consideration to provide optimal scope and right granular level of the business functionality in a service operation.

Normalization

Services are decomposed or consolidated to a level of normal form to minimize redundancy.

Optimization

High quality services are generally perferable to low quality services.

Relevance

Functionality is presented at a granularity recognized by the user as a meaningful service.

Encapsulation

Many services are consolidated for use under the SOA.

Location Transparency

This refers to the ability of a service consumer to invoke a service regardless of its actual location in the network.