

## Lecture 14 Service-oriented Architecture

- ❑ What is SOA
- ❑ SOA Reference Model
- ❑ SOA Analysis and Design (SOAD)
- ❑ SOA Modeling and Architecture (SOMA)

## What is SOA?

*Service-Oriented Architecture (SOA) is an architecture to*

- provide/consume services without knowing implementation of services or underlying platform/programming environment*
- services are published/discovered/requested through a formally defined interface that is independent to platforms*
- focus on integrating services with business processes a paradigm not bound to a particular platform or technology*

## SOA Definitions

[Open Group SOA Definition](#) (SOA-Definition)

[OASIS SOA Reference Model](#) (SOA-RM)

What Is Service-Oriented Architecture? ([XML.com](#))

What is Service-Oriented Architecture? ([Javaworld.com](#))

[Webopedia definition](#)

[TechEncyclopedia definition](#)

[OMG SOA Special Interest Group definition](#)

[WhatIs.com definition](#)

SearchWebServices.com

# Service-oriented Architecture (SOA)



## SOA/WS Principles

**Service Loose coupling** - Services maintain a relationship that minimizes dependencies

**Service contract** - Services are defined collectively by one or more service description documents

**Service abstraction** - Beyond what is described in the service contract, services hide logic from the outside world

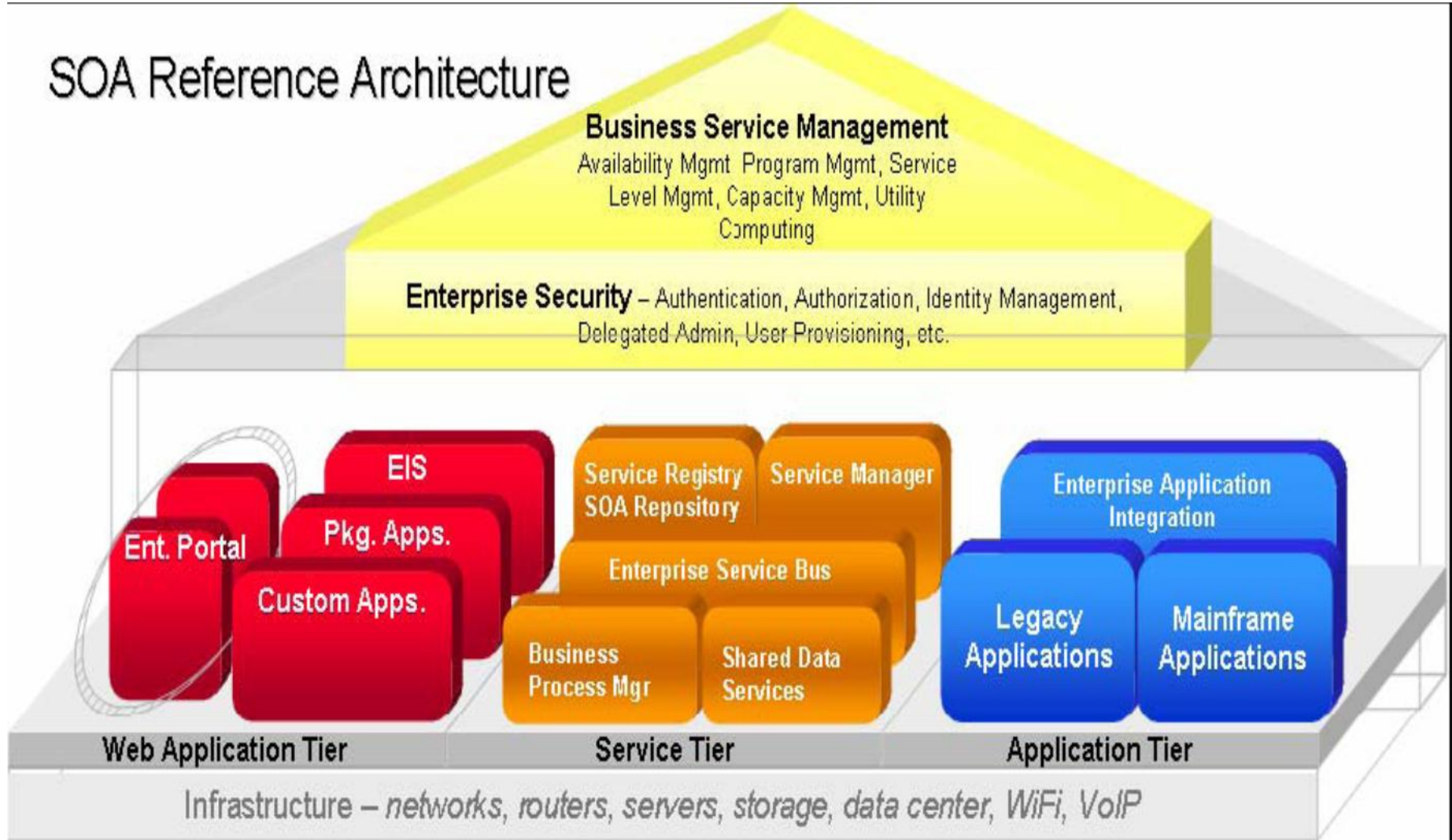
**Service reusability** - Logic is divided into services with the intention of promoting reuse

**Service autonomy** – Services have control over the logic they encapsulate

**Service statelessness** – Services minimize retaining information specific to an activity

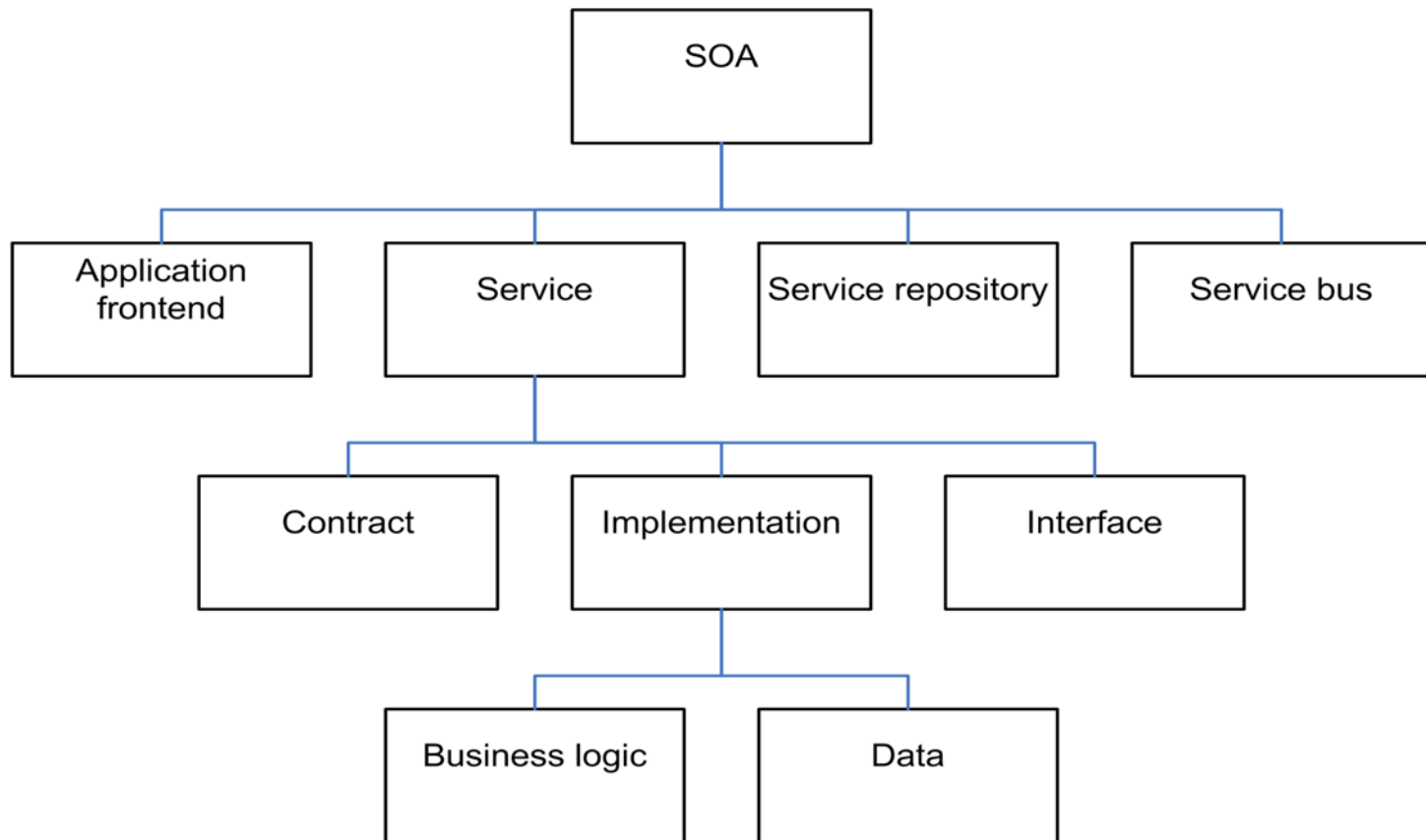
**Service discoverability** – Services are designed to be outwardly descriptive so that they can be found and assessed via available discovery mechanism

## SOA Reference Architecture



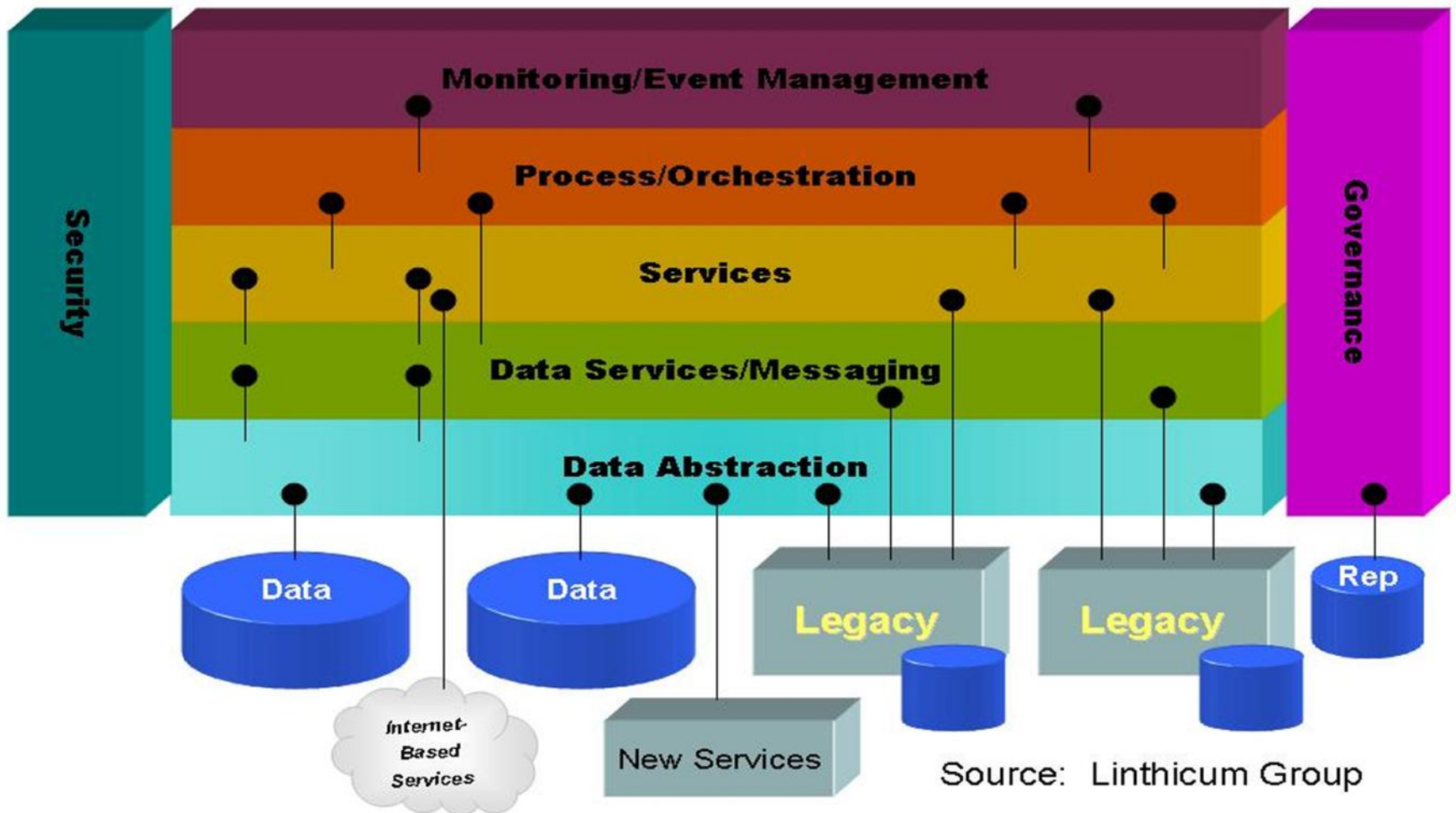


## SOA Elements



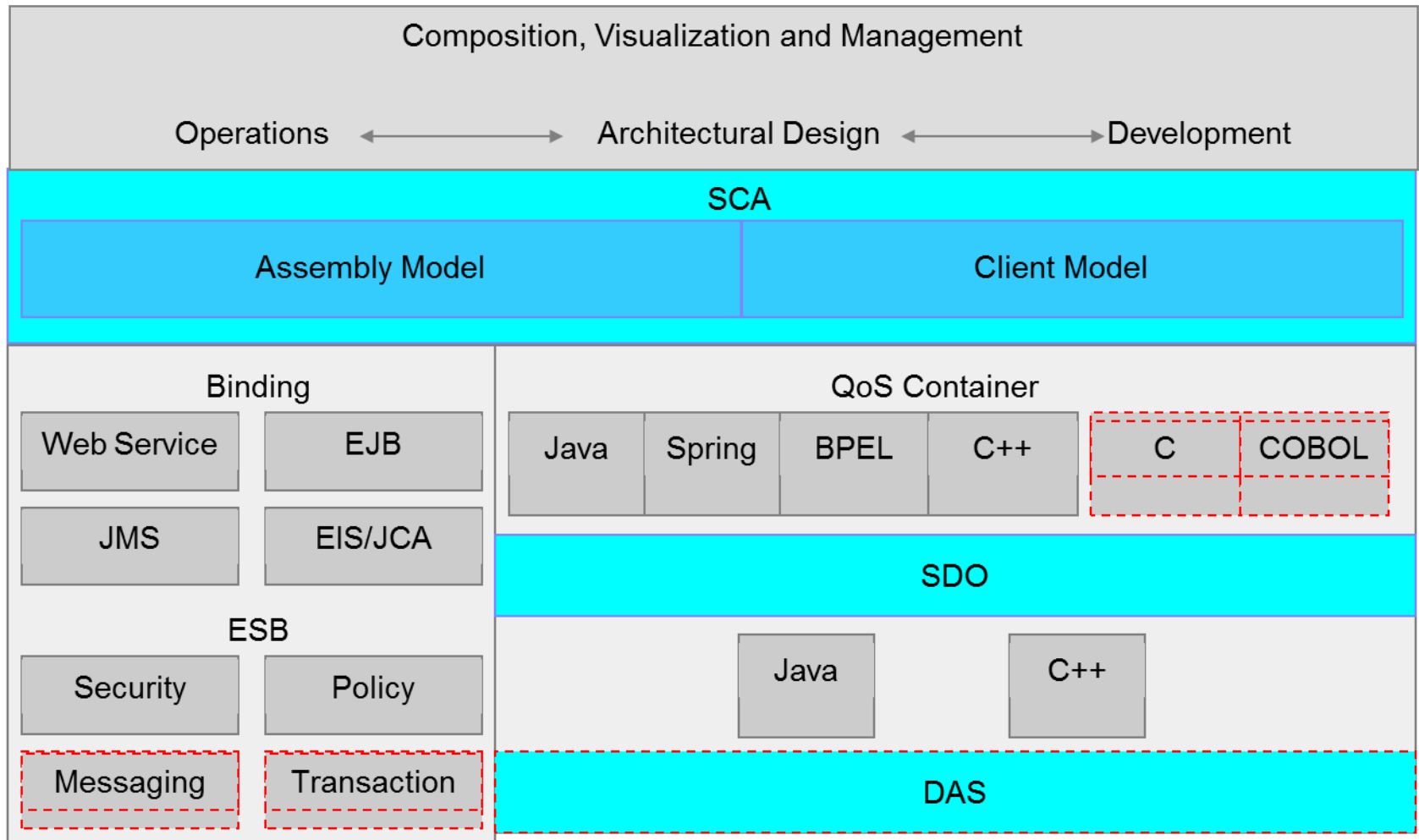
# Service-oriented Architecture (SOA)

SOA Meta-Model





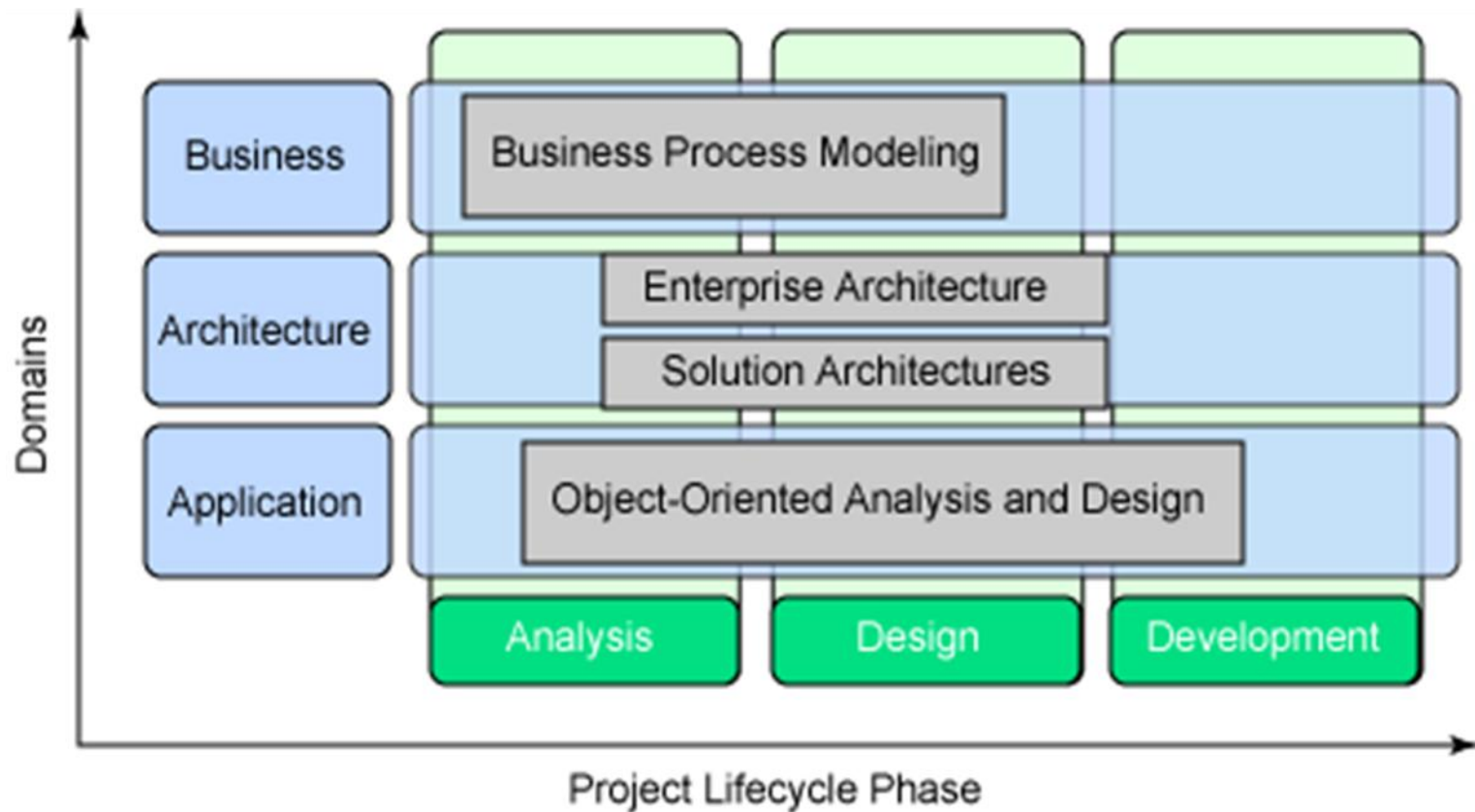
## OSOA规范构成



## Service-Oriented Methodology

- service-oriented analysis
- service-oriented design
- service-oriented analysis & design (SOAD)
- service-modeling
- service-oriented modeling
- service-oriented model & architecture (SOMA)
- business-driven development (BDD)

## Service-Oriented Analysis and Design



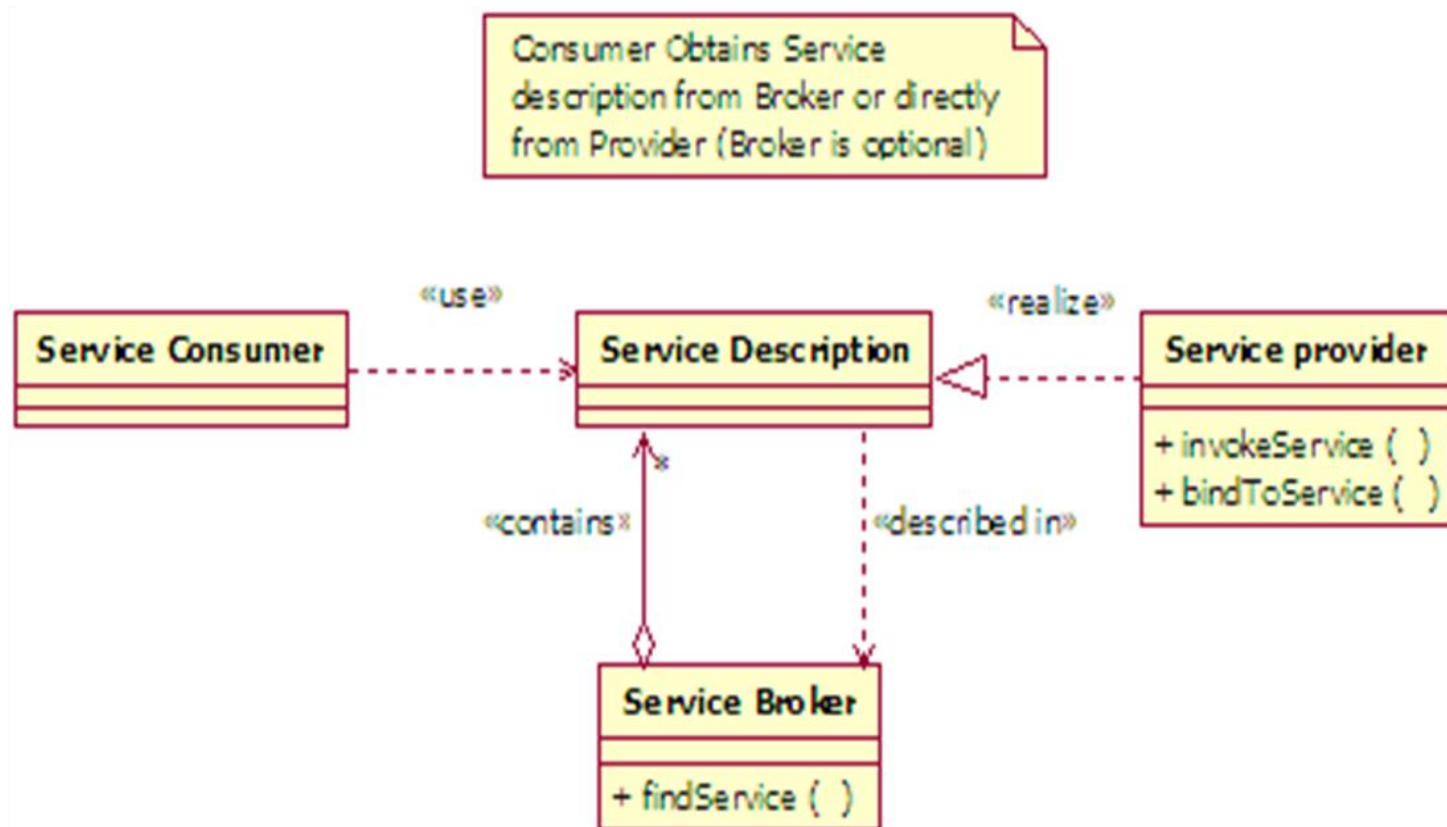
## Service-Oriented Analysis & Design (SOAD)

- Service identification and definition
- Domain decomposition
- Naming conventions
- Service granularity
- Business process and rules

## Service-Oriented Model & Architecture (SOMA)

- SOA meta-model
- SOA layer model
- SOA architecture document template

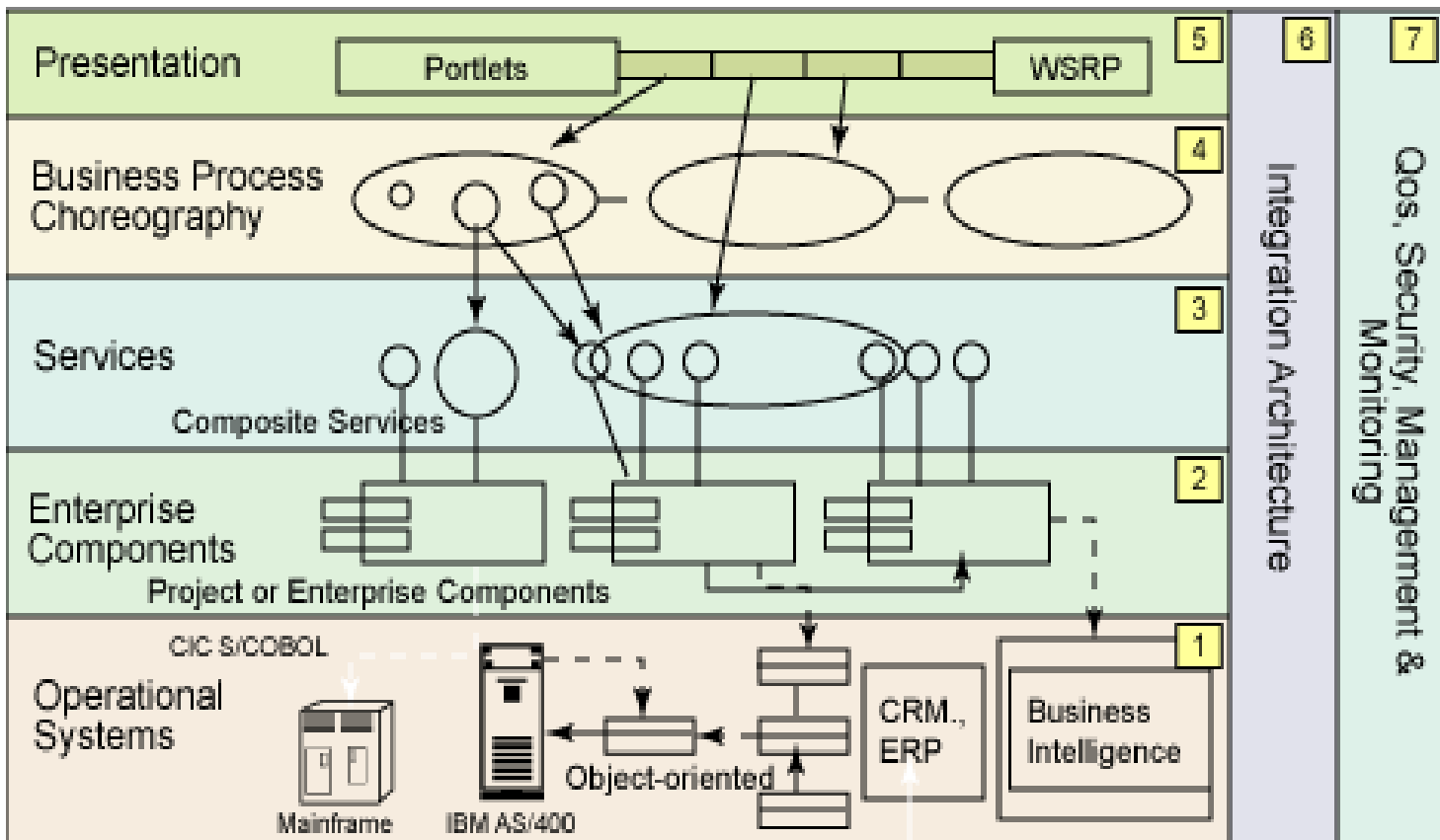
## SOA Meta-model - a conceptual model





## SOA Layer Model

- an abstract view of the layered architecture



## SOA architecture document template

Scope <what area of the enterprise is this architecture for?>

Operational systems layer

- Packaged applications

- Custom applications

- Architectural decisions

Enterprise components layer

- Functional areas

- business domains, goals and processes

- Decisions regarding governance

- Criteria

- Architectural decisions

## SOA architecture document template (cont'd)

### Business process and composition layer

- Business processes to be represented

- Architectural decisions

### Access or presentation layer

- Document implications of Web services and SOA

### Integration layer

- <Include considerations of an ESB>

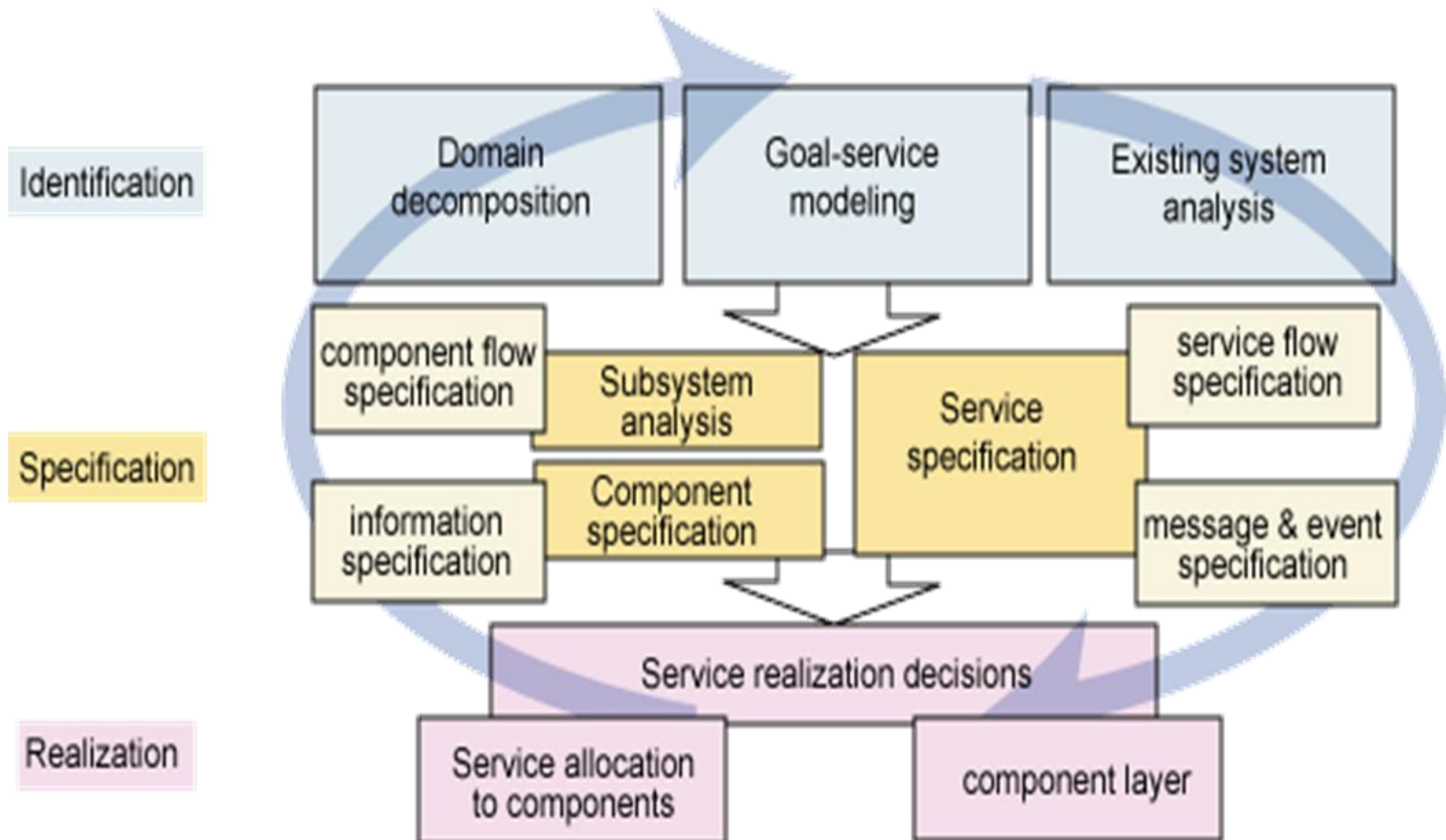
- Security issues and decisions

- Performance issues and decisions

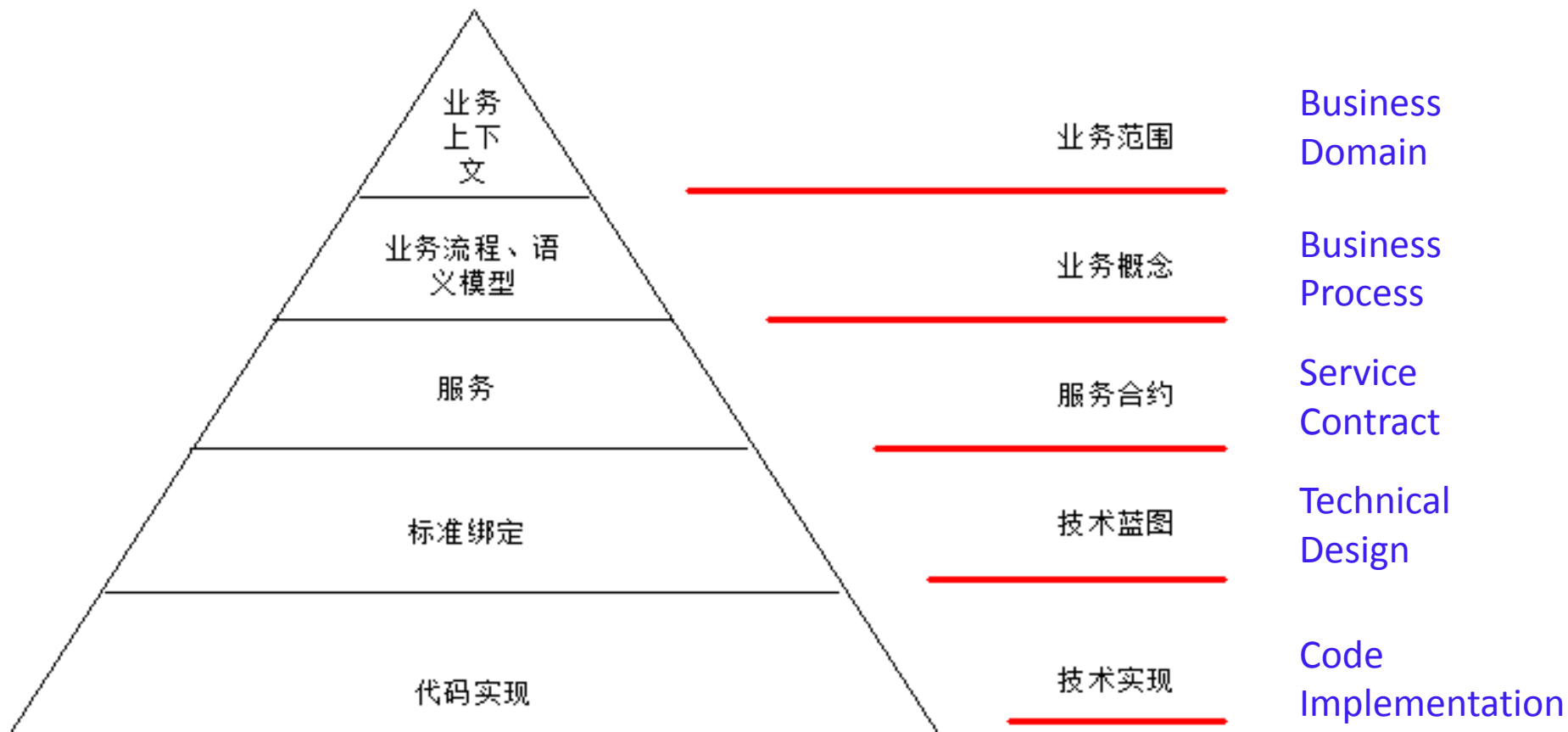
- Technology and standards limitations and decisions

- Monitoring and management of services

## SOA Modeling Method



## SOA Development Model



## SCA: Service Component Architecture

A Component-based Service-oriented Programming Framework, transferred to OASIS in 2007.

Two key functions:

- decoupling of components from transmission protocols
- decoupling of interface from programming languages



## SCA: Service Component Architecture (cont'd)

Key Concept: Service, which is independent to implementation technology

SCA provides service component model, assembly model and policy framework, which are well encapsulated and integrated.

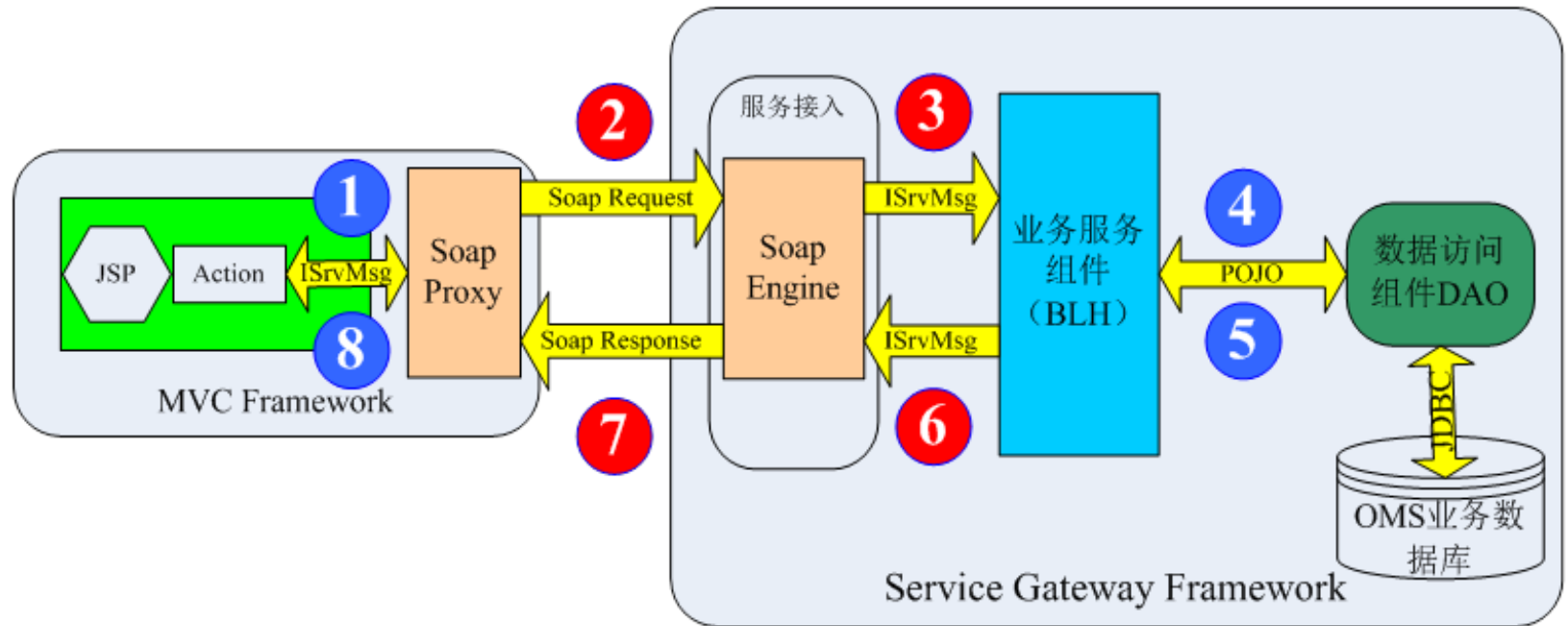
SCA Component: can publish services by using various protocols such as SOAP, RMI, REST, JMS or objective files of a virtual machine; can be implemented by using various types of technology, such as EJBs, Spring Beans, BPEL processes, COBOL, C++, Java, PHP, ...

Traditional component's functions are bind to transmission protocols, i.e. EJBs bind to RMI, Web Service bind to SOAP, etc., however, SCA decouples component services from transmission protocols and allows the implementation to select from different transmission protocols.

## SCA Key Elements

- Component, basic constructional unit and functional unit to provide services
- Service, business functions provided by the component, which can be deployed by other components
- Implementation, specifically refers to the code segments that implement the business functions
- Properties, a group of parameters that can be configured for specific services
- References, refers to the services provided by other components that are essential to implementation of this component's functions and services

## SOA Framework



Data  
units

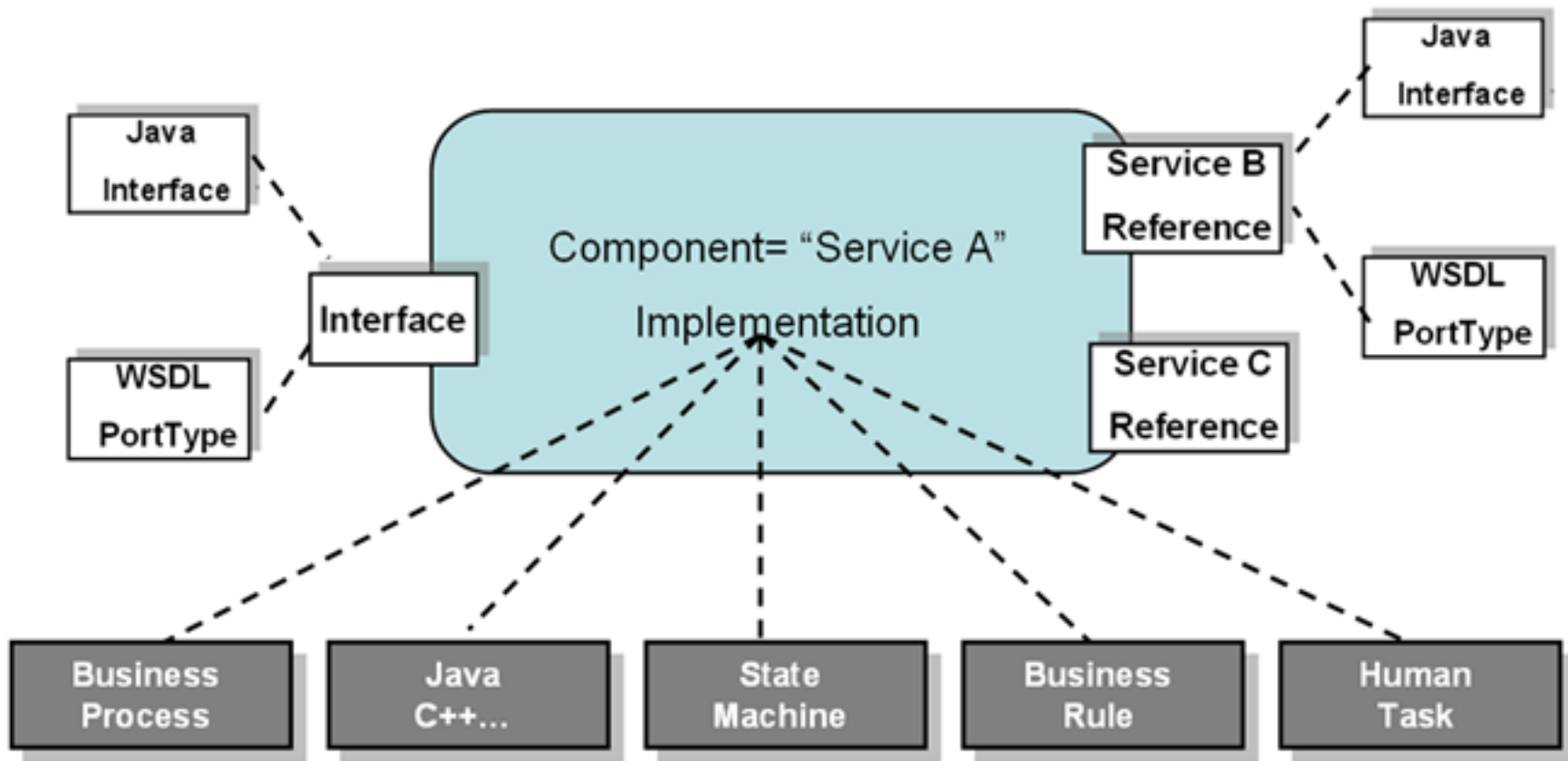
Data  
presentation

Service  
deployment

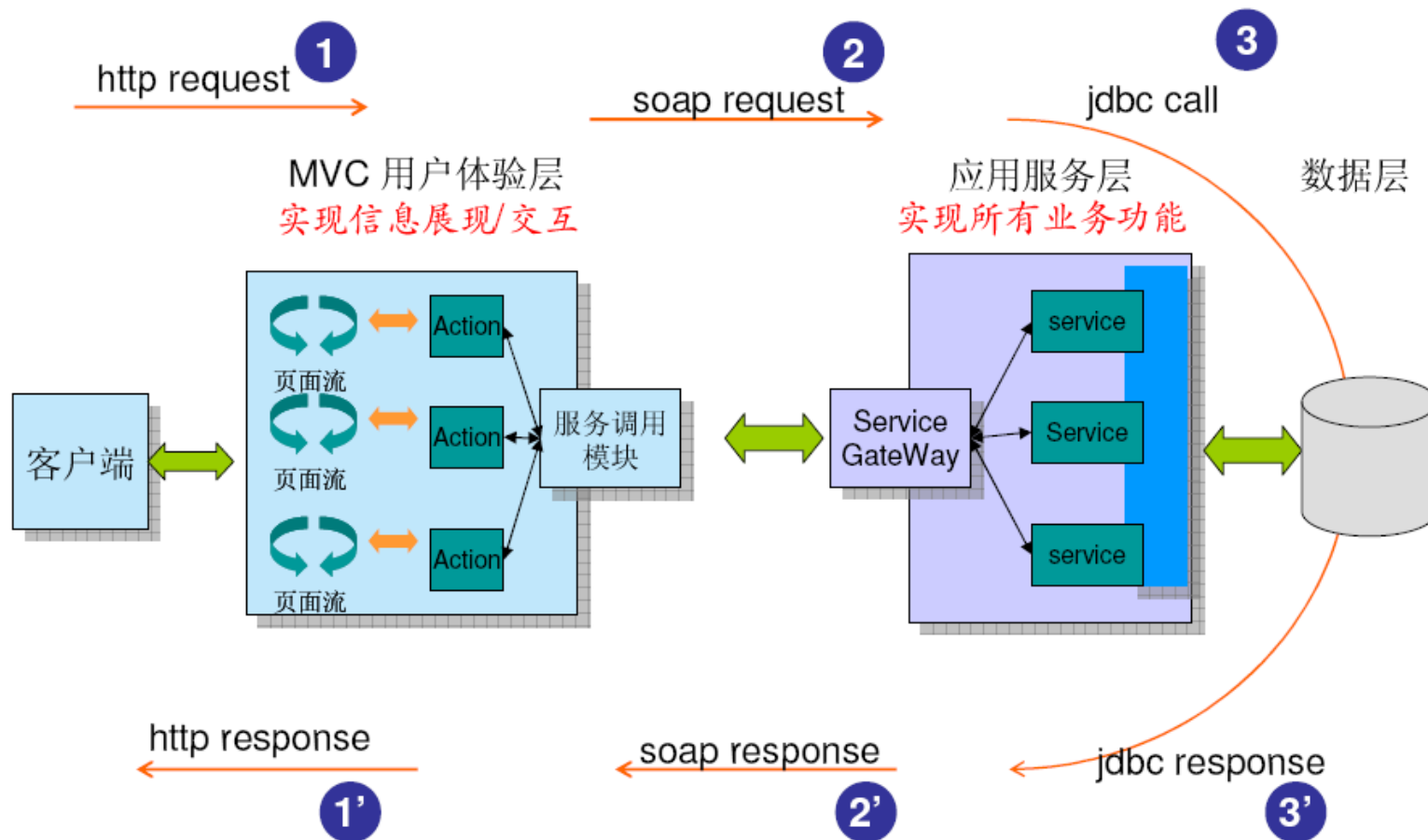
Business  
service

Data  
service

## SOA组件图



## SOA Data Flow



## SCA Specifications

- Assembly

Assembly is the process of composing business applications by configuring and connecting components that provide service implementations

- Client & Implementation

Whenever code based on another technology is used as part of an SCA module, the non-SCA code can use SCA services via the SCA client model.

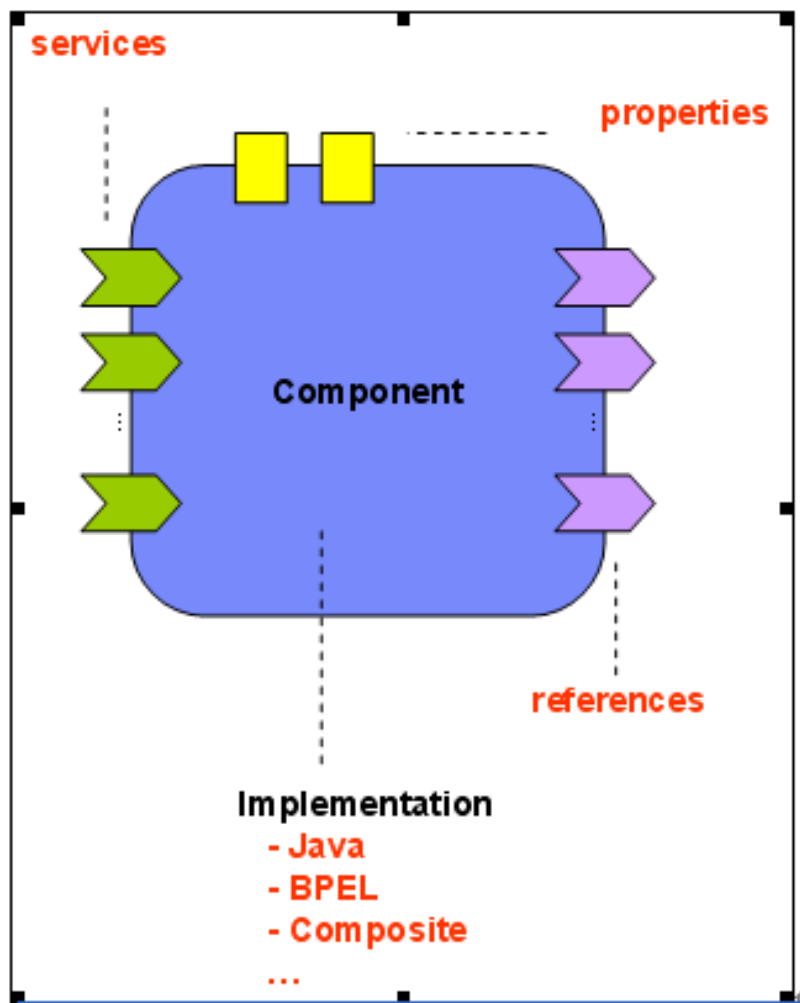
- Java
- C++
- BPEL
- Spring
- PHP

- Policy

- Security



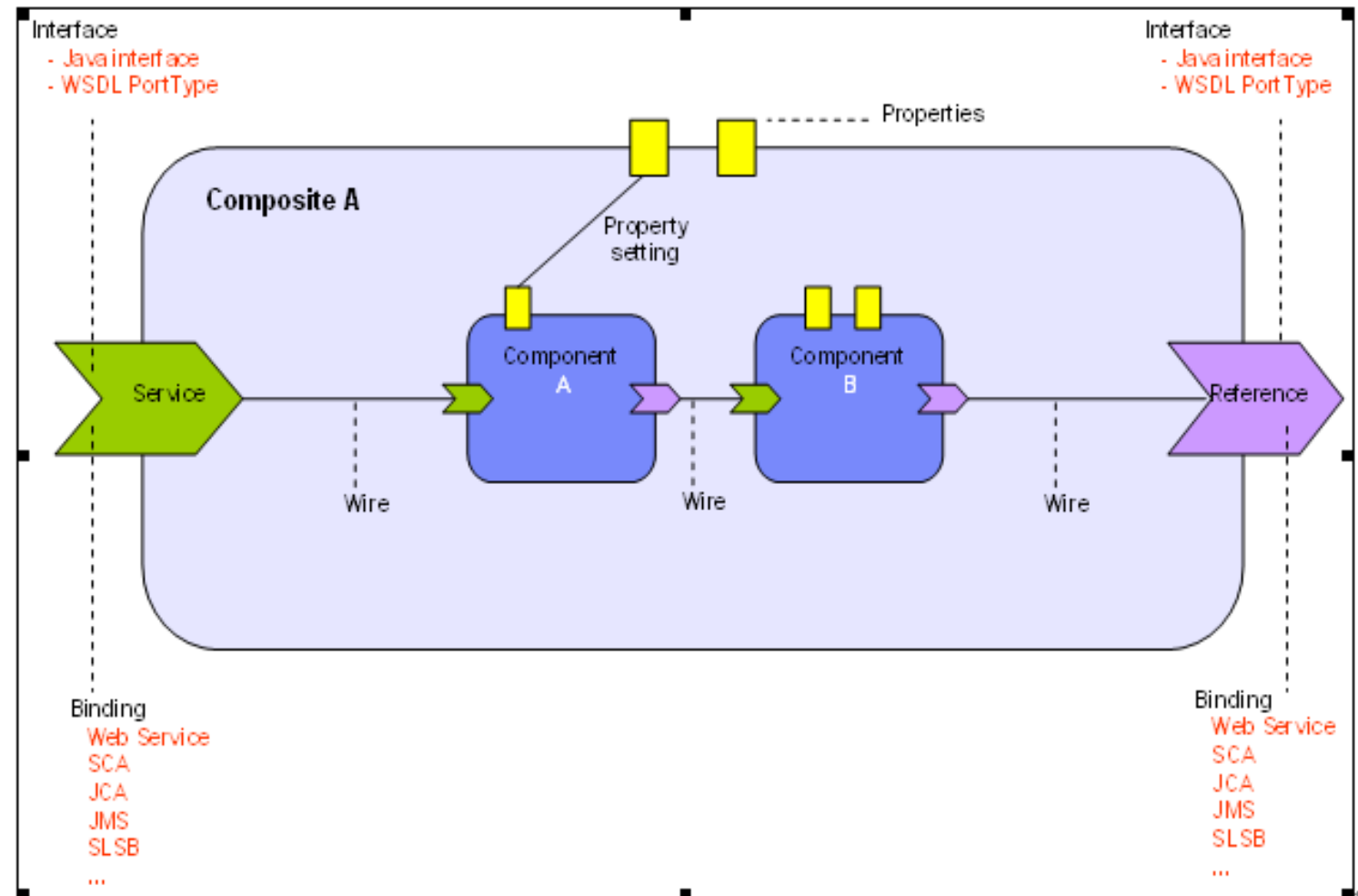
## SCA的Assemble模型（1）



## SCA Assembly : component

- Service
  - functions provided to others
- Interface
  - description to the port of a service (interface.java, interface.wsdl)
- Reference
  - needed services from others
  - description to the port of service needed from others
- Property
  - input parameters

## SCA Assemble Model



## SCA Assembly : component

```
<?xml version="1.0" encoding="ASCII"?>
<componentType xmlns="http://www.oesa.org/xmlns/sca/1.0"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema">

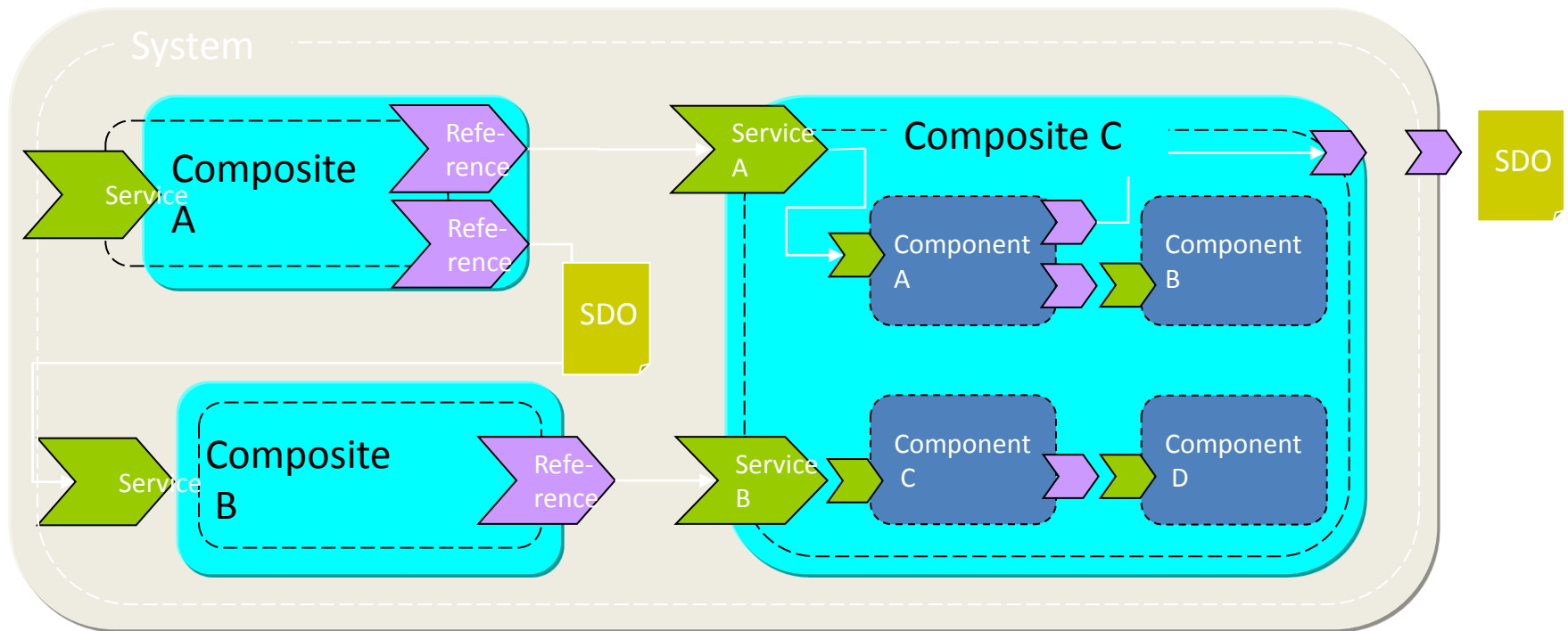
    <service name="AccountService">
        <interface.java interface="services.account.AccountService"/>
    </service>

    <reference name="accountDataService">
        <interface.java
        interface="services.accountdata.AccountDataService"/>
    </reference>

    <property name="currency" type="xsd:string">USD</property>

</componentType>
```

## SCA Assemble Model (cont'd)



## SCA Assembly : composite

```
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0"
            xmlns:eos="http://www.primeton.com/eos/6.0" name="account">
  <service name="AccountService" promote="AccountService">
    <interface.java interface="com.eos.bigbank.account.AccountService"/>
  </service>
  <component name="AccountService">
    <implementation.java
      class="com.eos.bigbank.account.AccountServiceImpl"/>
    <reference name="accountDataService"></reference>
    <property name="currency">USD</property>
  </component>
  <component name="accountDataService">
    <eos:implementation.eos
      component="com.eos.bigbank.account.AccountDataservice"/>
    <reference name=" AccountDataService">
      <binding.sca/>
    </reference>
  </component>
</composite>
```



## Advantages of SCA Framework

- Cross-platform and language independent. Allows to use various implementation technologies or different communication protocols for the application development
- Components are replaceable, which provides high flexibility in software assembly to meet various types of needs
- Services can be easily configured to be deployed
- Provide a specification of assembly, which supports a high efficient and fast assembly of components

End of Lecture

谢谢！