哈尔滨工业大学 国家示范性软件学院

第一章 概述

杨大易

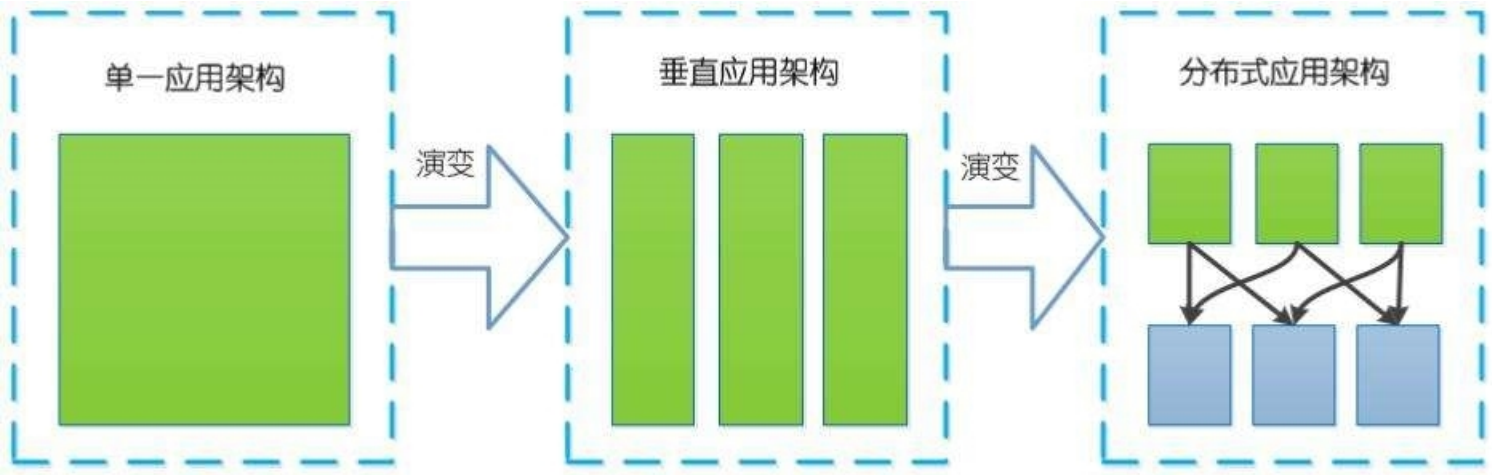
2023/2/10

本章内容

1. 面向服务软件系统的发展
2. 面向服务的基本概念
3. 面向服务软件系统实例
4. 面向服务有关技术

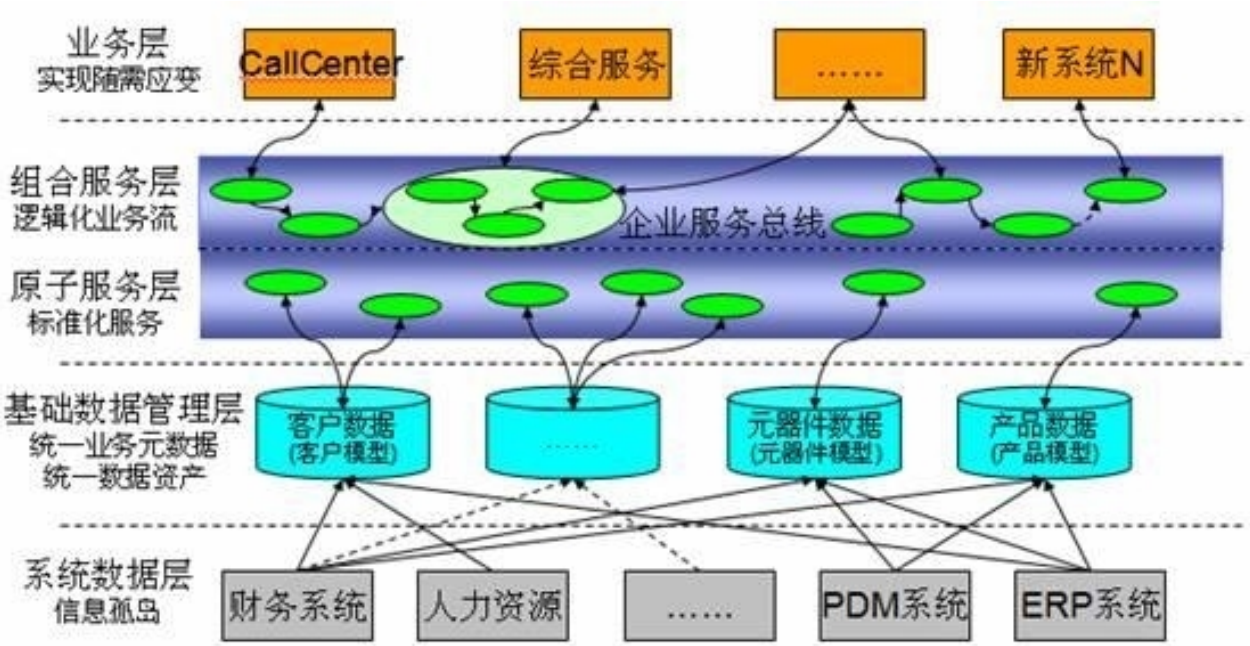
1.1 面向服务架构SOA

- 背景
  - 企业信息系统建设飞速发展。
  - 各部门构建各自系统，彼此功能和数据相互独立。
  - 需要与外部系统能够更加灵活的通信。



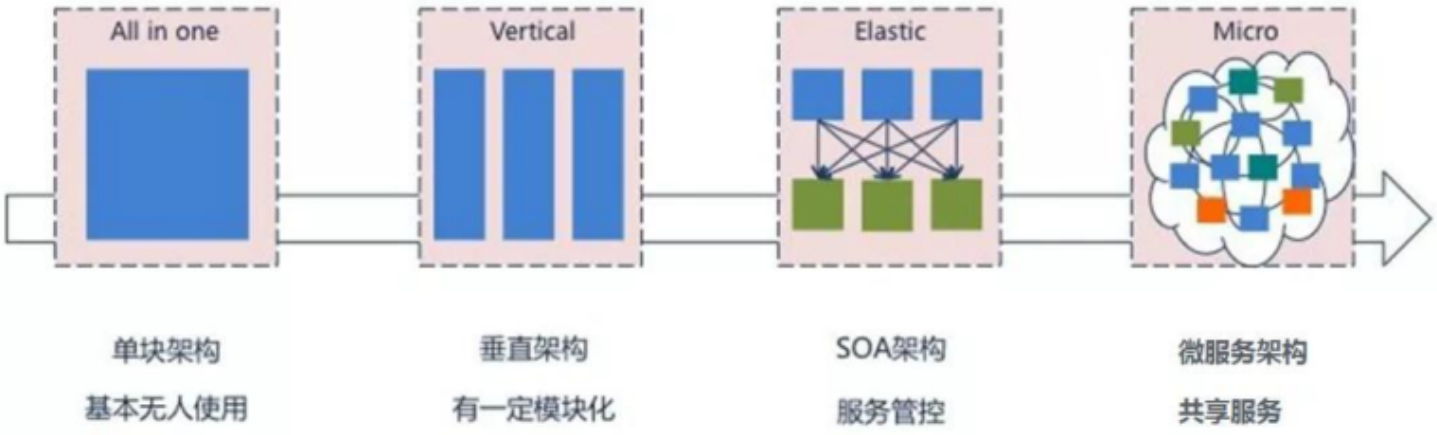
1.1 面向服务架构SOA

- SOA（面向服务架构）由Garnter在1994年提出
  - 把信息系统中需要整合的业务使用服务联系起来。
  - 解决企业级信息系统中“信息孤岛”、“重复造轮子”等问题。



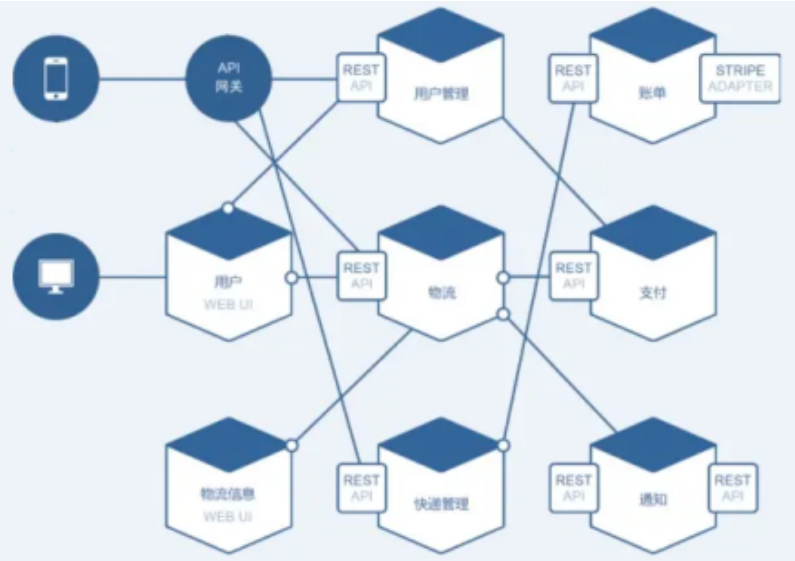
1.2 微服务技术

- 背景
  - 互联网快速发展，云服务广泛应用。
  - 云计算、大型机群系统、容器等技术日渐成熟。
  - 敏捷开发、领域驱动设计、持续交付等方法盛行。



1.2 微服务技术

- 微服务概念2012年出现，后因Martin Fowler流行
  - 围绕着业务领域组件来创建应用，这些应用可独立地进行开发、管理和加速。
  - 微服务架构使在云端部署、管理应用服务更加简单。



本章内容

1. 面向服务软件系统的发展过程
2. 面向服务的基本概念
3. 面向服务软件系统实例
4. 面向服务有关技术

2.1什么是Web Service

- ***The different functional units of an application or an enterprise, exists as a sing***
  applications and other ser
  message-based communic

```
<definitions name="StockQuote"
   targetNamespace="http://example.com/stockquote.wsdl"
             xmlns:tns="http://example.com/stockquote.wsdl"
             xmlns:xsd1="http://example.com/stockquote.xsd"
             xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
             xmlns="http://schemas.xmlsoap.org/wsdl/">
   <types>
      <schema targetNamespace="http://example.com/stockquote.xsd"
              xmlns="http://www.w3.org/2000/10/XMLSchema">
         <element name="TradePriceRequest">
            <complexType>
               <all>
                  <element name="tickerSymbol" type="string"/>
               </all>
            </complexType>
         </element>
         <element name="TradePrice">
            <complexType>
               <all>
                  <element name="price" type="float"/>
               </all>
            </complexType>
         </element>
      </schema>
   </types>
```

- 定义
  - 是应用程序组件
  - 使用开放协议进行通信
  - 独立的并可自我描述
  - 可被其他应用程序使用
  - **XML 是 Web Service 的基础**

2.2面向服务架构SOA

- *"An SOA is designed to provide the flexibility to treat elements of business processes and the underlying IT infrastructure as secure, standardized components (services) that can be reused to address changing business priorities."*

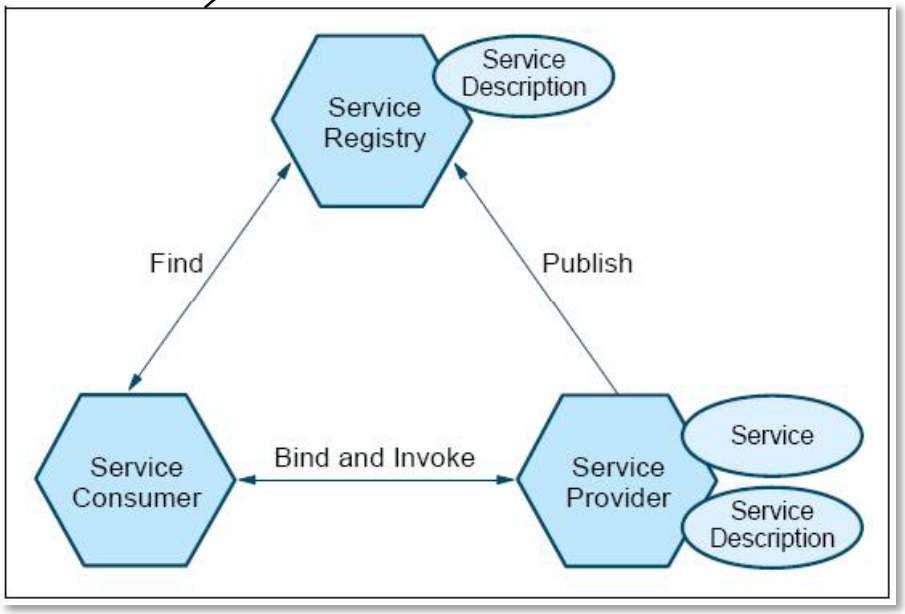
Source: http://www.ibm.com/soa

2.2面向服务架构SOA

- 面向服务的架构（SOA）是一个组件模型。
- 将应用程序的不同功能单元（称为服务）进行拆分，并通过这些服务之间定义良好的接口和契约联系起来。
- 接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。
- 使得构建在各种各样的系统中的服务可以以统一和通用的方式进行交互。

2.3 SOA的协作模式

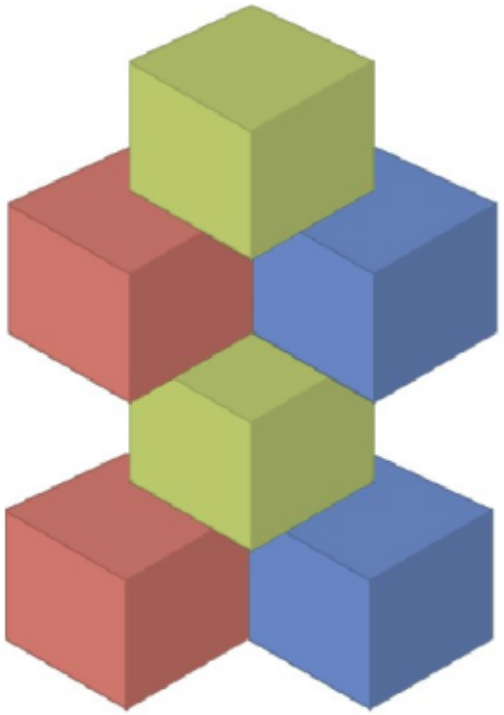
发布订阅式



- 角色
  - Provider：提供服务
  - Consumer：使用服务
  - Registry：注册服务
- 过程
  - Publish：发布服务
  - Find：查找服务
  - Bind and Invoke：绑定并调用服务

2.3 SOA的特征

- 可从外部访问
- 松散耦合
- 可重用的服务
- 服务接口设计管理
- 标准化的服务接口
- 支持各种消息模式
- 精确定义的服务契约

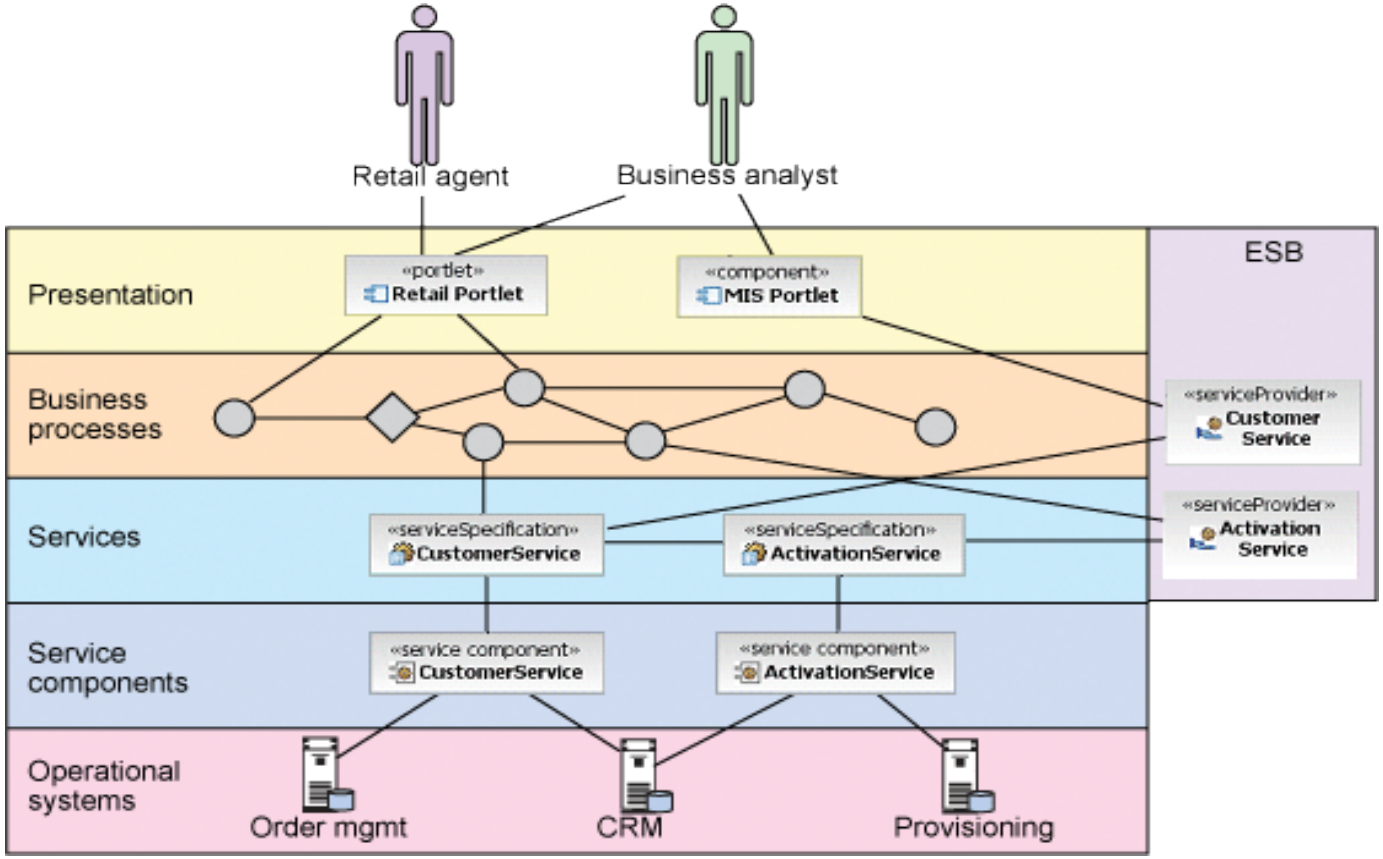


本章内容

1. 面向服务软件系统的发展过程
2. 面向服务的基本概念
3. 面向服务软件系统实例
4. 面向服务有关技术

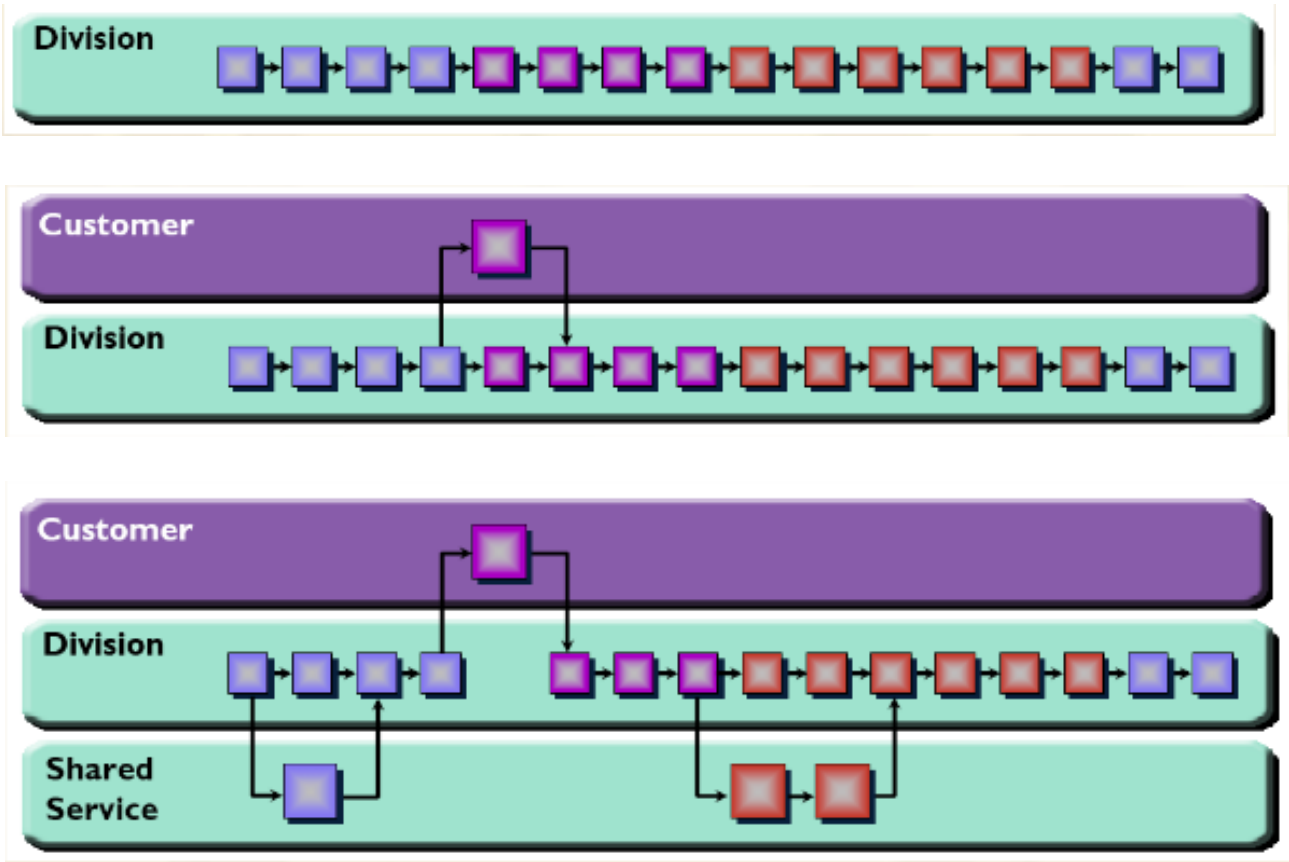
3.1 企业级应用系统

- 面向服务的企业级系统架构示例



3.1 企业级应用系统

- 灵活配置的业务流程



3.2 云服务


百度地图开放平台
功能与服务
解决方案
开发文档
反馈帮助
服务升级
NEW!
控制台
NEW!
Web服务API
概述
获取密钥
地点检索
地点输入提示
正/逆地理编码
轻量级轨迹服务
路线规划
批量算路
普通IP定位
智能硬件定位
鹰眼轨迹
实时路况查询
时区
批量服务
Web服务API
Web服务API
百度地图Web服务API为开发者提供http/https接口，即开发者通过http/https形式发起检索请求，获取返回json或xml格式的检索数据。用户可以基于此开发JavaScript、C#、C++、Java等语言的地图应用。
核心服务简介
北京市
GO
清华
北大
故宫
地点检索服务
支持城市、矩形及圆形区域关键字检索POI，返回json/xml格式的P...
清华大
GO
清华大学
清华大学地铁站
清华大学宿舍
地点输入提示服务
提供匹配用户输入关键字的辅助信息、提示接口、返回json/xml格...
lng
116.30
lat
40.05
address
北京市海淀区上地十街十号
POI
百度大厦附近
正/逆地理编码服务
通过地址获取坐标值或通过坐标点获取详细地址信息描述服务。
起
终
天坛
故宫
路线规划服务
支持公交、驾车、步行查询检索服务，返回json/xml格式的线路数...
time
60s
distance
600m
time
60s
distance
600m
time
60s
distance
600m
IP
61.135.169.81
City
北京
两条轨迹重合率 =76.4%

3.3 微服务架构系统

- 阿里核心微服务技术架构

本章内容

1. 面向服务软件系统的发展过程
2. 面向服务的基本概念
3. 面向服务软件系统实例
4. 面向服务有关技术

4.1 较早的分布式架构

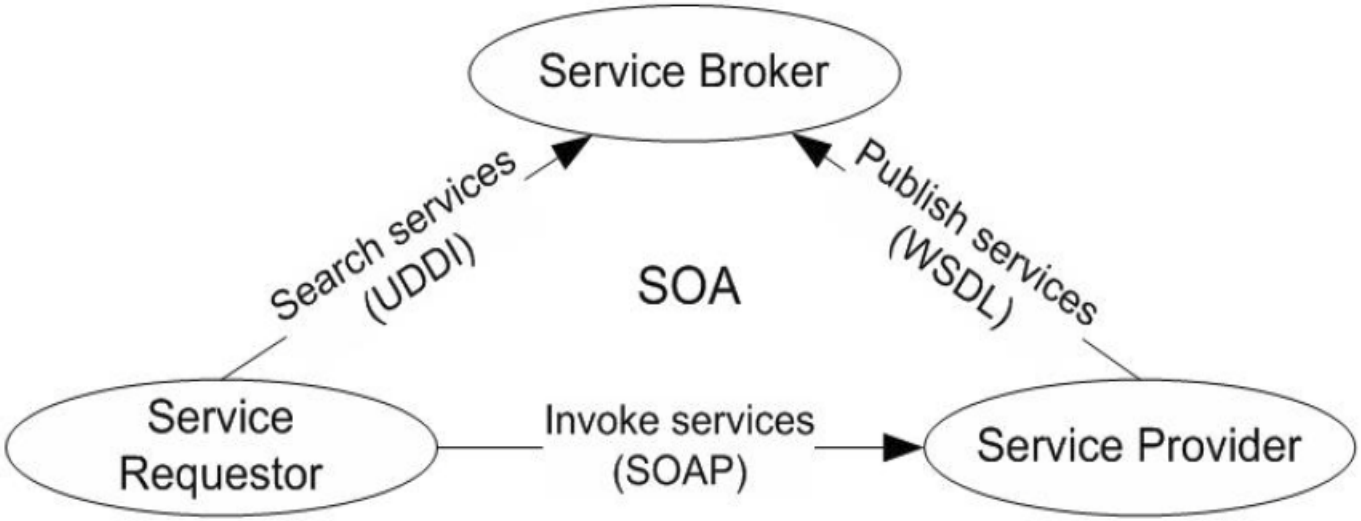
- **DCOM**
  - 分布式组件对象模型
  - 一系列微软的概念和程序接口，客户端程序对象能够请求来自网络中另一台计算机上的服务器程序对象

兼容性差，跨平台困难

- **CORBA**
  - 公共对象请求代理体系结构
  - OMG组织制订的标准的面向对象应用程序体系规范

4.2 分布式服务

- **Web Service**
  - **XML：Web Service 技术的基础**



4.2 分布式服务

- **Restful： Representational State Transfer**
  - 首次出现在 2000 年 Roy Fielding 的博士论文中。
  - 一种软件架构风格、设计风格，而不是标准，只是提供了一组设计原则和约束条件。
  - 客户端和服务器之间的交互在请求之间是无状态的。
  - 服务器端的应用程序状态和功能可以分为各种资源，每个资源都使用 URI (Universal Resource Identifier) 得到一个唯一的地址，使用的是标准的 HTTP 方法。
  - Restful相比于SOAP以及XML-RPC更加简单明了。

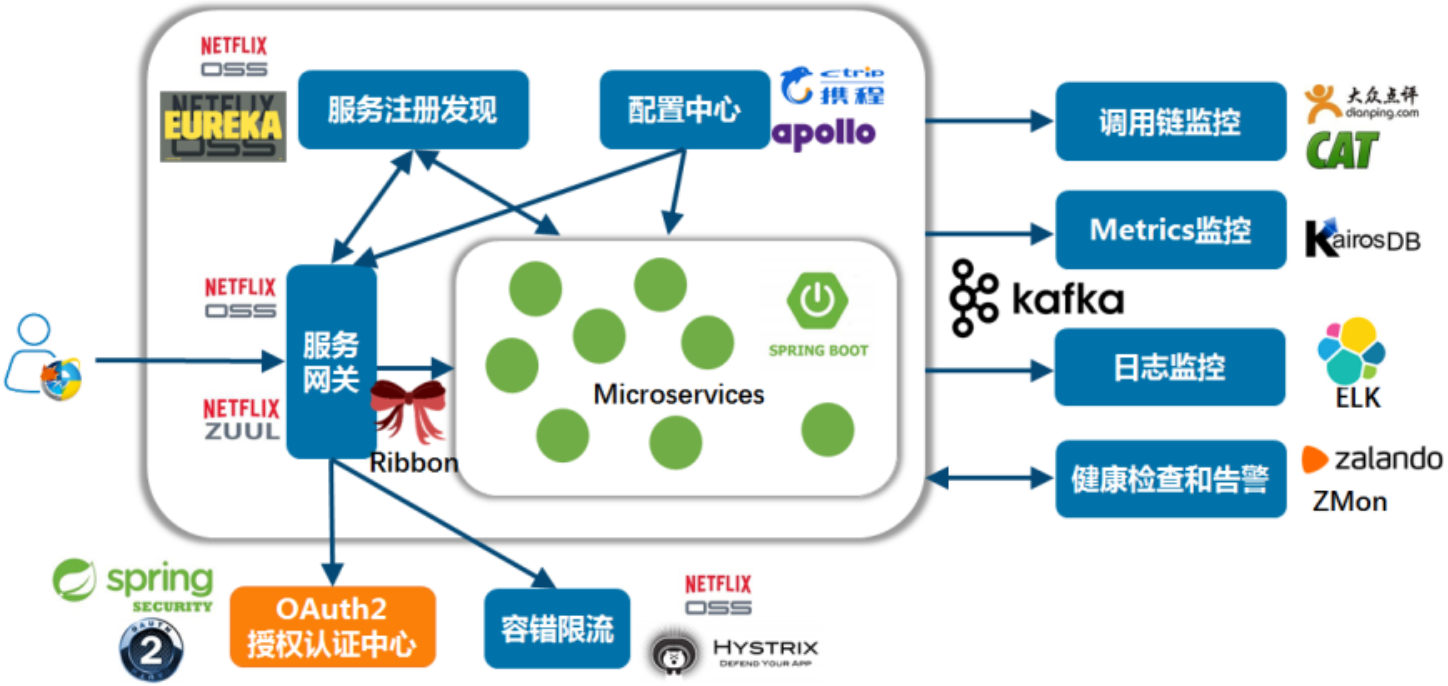
4.3 企业级服务架构

- Web Service
- ESB：Enterprise Service Bus
  - 提供了分布式的运行管理机制，支持基于内容的路由和过滤，具备了复杂数据的传输能力，并可以提供一系列的标准接口。
- BPEL：Business Process Execution Language
  - 基于XML的，用来描写业务过程的编程语言，被描写的业务过程的每个单一步骤则由Web服务来实现。

4.4 微服务

Spring Cloud框架

- 组件方案实例



4.4 微服务

- Dubbo：开源Java RPC框架。远程方法调用，智能容错和负载均衡，服务自动注册和发现
- Zookeeper：开源分布式服务治理
- Docker：开源的应用容器引擎
- Spring Boot：Java平台开源框架，简化Spring应用的搭建和开发过程
- Jenkins：持续的软件版本发布及测试服务
- 敏捷开发

本章小结

- SOA和微服务架构是当前软件系统设计的重要架构方案。
- 与面向服务有关的技术与工具繁多，是当前行业主流技术方向。

谢谢！