# 我所理解的SOA和微服务

SOA和微服务到底是什么关系？

说实话，我确实不明白SOA和微服务到底有什么本质上的区别，两者说到底都是对外提供接口的一种架构设计方式。我倒觉得微服务其实就是随着互联网的发展，复杂的平台、业务的出现，导致SOA架构向更细粒度、更通过化程度发展，就成了所谓的微服务了。以这种说法做为根据，我觉得SOA与微服务的区别在于如下几个方面：

1. 微服务相比于SOA更加精细，微服务更多的以独立的进程的方式存在，互相之间并无影响；
2. 微服务提供的接口方式更加通用化，例如HTTP RESTful方式，各种终端都可以调用，无关语言、平台限制；
3. 微服务更倾向于分布式去中心化的部署方式，在互联网业务场景下更适合；

为什么要使用微服务？

技术为业务而生，架构也为业务而出现，当然SOA和微服务也是因为业务的发展而出现。出现SOA和微服务框架与业务的发展、平台的壮大密不可分，下面借用dubbo的网站架构发展图和说明：



* **单一应用架构**
  + 当网站流量很小时，只需一个应用，将所有功能都部署在一起，以减少部署节点和成本。
  + 此时，用于简化增删改查工作量的 **数据访问框架(ORM)** 是关键。
* **垂直应用架构**
  + 当访问量逐渐增大，单一应用增加机器带来的加速度越来越小，将应用拆成互不相干的几个应用，以提升效率。
  + 此时，用于加速前端页面开发的 **Web框架(MVC)** 是关键。
* **分布式服务架构**
  + 当垂直应用越来越多，应用之间交互不可避免，将核心业务抽取出来，作为独立的服务，逐渐形成稳定的服务中心，使前端应用能更快速的响应多变的市场需求。
  + 此时，用于提高业务复用及整合的 **分布式服务框架(RPC)** 是关键。
* **流动计算架构**
  + 当服务越来越多，容量的评估，小服务资源的浪费等问题逐渐显现，此时需增加一个调度中心基于访问压力实时管理集群容量，提高集群利用率。
  + 此时，用于提高机器利用率的 **资源调度和治理中心(SOA)** 是关键。

平台随着业务的发展从 All in One 环境就可以满足业务需求（以Java来说，可能只是一两个war包就解决了）；发展到需要拆分多个应用，并且采用MVC的方式分离前后端，加快开发效率；在发展到服务越来越多，不得不将一些核心或共用的服务拆分出来，其实发展到此阶段，如果服务拆分的足够精细，并且独立运行，我觉得就可以将之理解为一个微服务了。

理想中的微服务架构

没有什么东西是完美的，网站架构也是这样的，只有「比之前好一点」的架构或「目前最好的实现方式」，不存在理想中的架构，那么理想中微服务架构应该是怎么样的呢，我觉得至少应该有如下几个特点：

1. 能支持当前业务需求，当然这只是最最基本的条件；
2. 每个微服务都要去中心化，不存在单点故障；
3. 每个微服务都要实现高可用、高负载，不会因为一个服务不可用而影响了整套业务流；
4. 每个微服务都要高度通用化，即多种终端都可调用，不分语言和平台；
5. 服务部署或升级简单，不会消耗大量人力并且部署过程不易出现人为错误；
6. 微服务具有快速注册与自动发现功能（例如dubbo框架）

当然，这只是其中能想到的几点，实际环境中用到的微服务框架有可能会根据实际业务需求优化出更加个性化的功能，也可能有些功能是不需要的。还是那句话，架构是服务于业务的，能快速方便的满足业务需求的架构才是好的架构，才是好的微服务架构。