# 论SOA技术的应用

## 概要：

本人于2016年03月参加了秦皇岛某国际货运物流公司的物流管理系统开发，在该项目中担任架构师工作，主要负责该系统架构和网络安全体系的架构设计。该系统包含物流过程所涉及的各个领域的信息。包括销售、运输、仓储、海关、码头、堆场等。随着公司物流吞吐量不断增大，为满足高性能，高效率的要求开发此系统。

为了解决该系统中复杂，分散，异构的数据信息之间的交换，实现数据的高可用性，同时适应新的业务需求，开发新的应用系统适应日益增长的港口吞吐量与订单管理的需要，实现平台的以扩张性，易集成的特性，我们采用WebService开发技术。建立SOA架构，实现了系统的高可复用性。本文以物流管理系统为例，描述SOA技术在项目中的应用过程以及使用效果。本文最后，讨论SOA技术的有关想法和教训。

本人在2019年10月参加了杭州某互联网公司的二手商品交易系统的开发，在该系统开发中本人担任架构师的工作。由于本公司采用了C2C的经营模式，所以此业务发展空间大，所以需要具备一定的扩展性，并且要保证一定的高可用。该系统包含了用户系统，购物车系统，订单系统，库存系统，搜索系统，即时通信系统，物流系统。

由于上级部门对该系统的并发量提出了较高的要求，且很多模块的复用性较强

（2.1）本人于2019年10月参加了杭州某互联网公司的二手商品交易系统开发，并在该系统中担任架构师的工作。该系统包含用户系统，购物车系统，订单系统，库存系统，搜索系统，即时通信系统，物流系统。 此系统需要在日后电商活动的高并发量、高访问量的场景里克服通信阻塞，服务雪崩，缓存击穿等一系列由于负载过大而造成的种种故障，并保证较低响应时间，较高的可用性。我们需要使用较为轻量级的通信方法、服务降级、服务熔断、负载均衡、缓存等技术。由于我们的运维采用了第三方云服务提供商，在活动结束时，需要及时释放资源，将资源给公司的其他业务运行，所以我们需要松耦合，可插拔的服务设计。基于以上考虑，我们采用了微服务的架构。

本人于2016年03月参加了秦皇岛某国际货运公司物流管理系统的开发。该系统包含物流过程所涉及的各个领域的信息系统，包括销售、运输、仓储、海关、码头、堆场等。随着公司业务发展，国内国际物流吞吐量不断增大，为了满足高性能，高效率的要求开发物流管理系统。该项目由于工程较大，共分三期完成。我主要负责二期与三期工程，在该项目中担任系统架构师工作，主要负责该系统架构和网络安全体系架构设计。

本人于2019年10月参加了杭州某互联网公司的二手商品交易系统开发，并在该项目中担任架构师的工作。该系统包含二手交易过程中所需要的各个信息系统，包括用户身份，购物车，订单，库存，物流，用户喜好等。考虑到公司采用了C2C的经营模式，此业务发展空间较大，后期用户增多较快，订单成交量较多，所以此系统需要具备易扩展性、易集成性、高性能。

该二手交易系统（以下简称STS系统），需要在日后电商活动的高并发量、高访问量的场景里最大程度避免宕机，服务雪崩，缓存击穿等一系列由于负载过大而造成的种种故障，并保证较低响应时间，较高的可用性。我们需要使用较为轻量级的通信机制、服务降级、服务熔断、负载均衡、缓存等技术。由于我们的运维采用了第三方云服务提供商，在活动结束时，需要及时释放资源，将资源给公司的其他业务运行，所以我们需要松耦合，可插拔，支持独立部署的系统设计。基于以上考虑，我们采用了微服务的架构。

我们采用微服务架构，是因为微服务在系统开发好电商场景下具有种种优势。微服务使用较为轻量级的通讯机制，使得服务之间的通讯较为简单，其异步通信机制能缓解通信阻塞。微服务颗粒度较小，能提高组件的复用，简化了开发人员的开发，能够很好适应敏捷开发。微服务支持独立的部署，本系统较为复杂，构件的独立部署能方便整个项目的部署，并且每个构件能独立扩展。而且在一线场景中，能够限制一些构件的计算量，让这些构件仅做到柔性可用，让一些重要的构件获得大量的资源，提高整个系统的吞吐率和响应速度，让用户有良好的购物体验。此项目比较大，使用微服务能够细化团队，从而获得理想的团队及生产力。

该物流管理系统（以下简称SMMED系统）包含物流过程所涉及的各个领域。他由计算机、应用软件以及其他高科技的设备，通过全球通信网络连接起来的，立体的，动态互动的系统。为了解决系统中复杂、分散、异构的数据信息之间的交换，实现数据高可用性，同时适应新的需求。实现平台的易扩张性、易集成性，我们分析决定采用WebService开发技术，建立SOA架构。SOA是一种粗粒度，松耦合的服务架构。服务之间通过精确定X接口进行通信，不涉及底层接口和通信模型。

分析原有一期工程发现，原有系统服务设计接口不规范，且消息模式较为单一，针对此问题，我们与团队首先对一期工程接口进行了部分修改设计，并对二期工程需要完成的功能进行拆分，我们根据具体功能，将SMMED系统拆分为：物料需求系统（MRPS），制造资源系统（MRPIS），企业资源系统（ERS）以及分销资源系统（DRS）四个子系统。具体逻辑架构如下：

接下来就SOA系统中部分应用过程进行详细说明：

SOA管理中心：SOA管理中心，这是该架构的核心部分，主要功能如下：1，提供发布/订购/ServiceAdapter组件。2，对服务访问的各种配置。3，对服务访问的授权描述。4，对服务出现安全问题进行通知与报警。

每个系统在需要服务时，去订阅服务，然后拉取服务地址MyNeedServiceList,然后通过ServiceAdapter访问服务，Filter会做权限校验及服务被调用的信息采集。然后在管理中心添加服务，文档描述。各个系统之间是直接交互的，服务调度不走中转路由，组件的作用仅是辅助性的约束，达到了低侵入式的要求，后期维护成本低。

由于侵入性次小，所以各个系统之间的服务变更维护完全由各自研发团队维护。本系统之间通讯不走服务，直接内部调用。调用通过ServiceAdapter组件访问。ServiceAdapter包含对进程内WCF,WebAPI等访问的封装，这样便于以后替换为其他服务，各服务在拓展上不受管理中心限制，自行做负载，增加服务器即可。

2.聚合服务

服务的颗粒度一直是SOA设计的关注点，粒度太粗则难以复用，粒度过细则需要多次往返交互，其性能，事务处理都会受到影响，如在DRS系统中订单服务，这个过程包含创建用户资料，生成预订单，重新账务关系，更新库存等一系列操作。这是个粗粒度的服务，里面包含若干子系统提供的粗粒度服务，细粒度服务下，多个子系统避免不了互相交互，长久下去会让系统过于沉重，变得职责混乱。

服务设计准则就是让服务高内聚，服务之间松耦合，边界清晰。所以我们抽离一个聚合服务系统，它专门负责把各系统提供的细粒度服务进行整合，提供给前端使用，而其他各个系统只做自己服务之内的事情，在聚合服务中，方便我们更合理把控服务的颗粒度，提高服务重用。

3．服务分级：  
 多个研发团队协作时，很难根据个人对全部业务熟悉。所以为了避免服务调用混乱，甚至循环依赖调用，增加了服务的分级。

按图中所示，1级服务不能调用2级服务，即低级不能调用高级，高级可调用低级。同级互相调用，这个级别针对单个服务而分，比如更新库存服务，它没有外部依赖服务，只是更新自己的DB，这样我们就将其设为一级服务。而订单服务，它内部调用用户资料DB，订单DB，账务DB一级库存DB，那么我们就将其定位于2级服务。

通过使用SOA架构，系统的架构更加清晰，业务提供者和业务服务使用者的松散耦合关系及时开放标准的采用确保了其易维护特性的实现，建立在以SOA基础上的系统。当需求发生变化的时候，不需要修改提供业务服务的接口，只需要调整业务服务流程或者修改操作即可，整个应用系统也更容易被维护，使用者无需了解提供者的具体实现细节，具有更高的可用性；依靠业务服务设计，开发和部署等采用的架构模型实现伸缩性，使得服务提供者可以互相彼此独立地进行了调整，以满足新的需求。

通过综合利用以上设计，系统于2018年12月投入了运行，并取得了较好的反馈，公司业务流程更加清晰，业务水平明显提高。系统性能得到了公司的好评。

尽管系统建设取得了成功，但是我也看到了不足之处，设计采用SOA技术，虽满足了公司对性能的要求，但在业务集成过程中，部署过程想当繁琐，多个系统对服务器有着不同的性能要求，在设计之初未考虑其迁移性，导致集成迁移过程进行了较长时间。

如果未来对该系统进行修改，希望能与团队来用更好的技术提高其可移植性