**摘要:** 【问题】 系统架构风格（System Architecture Style）是描述某一特定应用领域中系统组织方式的惯用模式。架构风格定义了一个词汇表和一组约束，词汇表中包含一些构件和连接件类型，而这组约束指出系统是如何将这些构件和连接件组合起来的。软件系统架构风格反映了领域中众多软件系统所共有的结构和语义特性，并指导如何将各个模块和子系统有效地组织成一个完整的系统。软件系统架构风格的共有部分可以使得不同系统共享同一个实现代码，系统能够按照常用的、规范化的方式来组织，便于不同设计者很容易地理解系统架构。 请以“软件系统架构风格”论题，依次从以下三个方面进行论述： 1．概要叙述你参与分析和开发的软件系统开发项目以及你所担任的主要工作。 2．分析软件系统开发中常用的软件系统架构风格有哪些？详细阐述每种风格的具体含义。 3．详细说明在你所参与的软件系统开发项目中，采用了哪种软件系统架构风格，具体实施效果如何。 【摘要】 2019年10月，本人所在保险公司启动了超级销售APP项目，该项目通过运用先进的销售工具、客户管理、营销活动管理等功能以达到提升销售人员的效能，加大业务驱动的目标。该项目中我担任系统架构师，负责系统的架构设计工作。本文以该项目为例，主要论述在系统架构风格中的具体应用。通过微服务风格，实现业务功能拆分为出单、业绩、客户、营销、续保等微服务，服务与服务采用轻量级的通信机制，每个服务围绕着具体业务进行构建，解决传统服务架构功能集中应用复杂问题；通过虚拟机风格，实现营销活动规则配置，动态生成如拉力赛、续保专项提升等激励活动；通过基于服务的架构风格，实现微服务间通信服务注册，与周边系统集成采用企业服务总线。基于以上技术的应用，项目成功上线。

本人所在的保险公司分支机构遍布全国，已设立分公司36家，机构总数超过2100家，营业机构覆盖全国各个省份，系统员工人数超6万人。因保险生态体系的变革，各保险公司都在积极科技转型，公司基于新业态发展通过“线上化、数字化、智能化”加速推进“三新三聚焦”的战略转型。故启动了超级销售APP项目建设，本项目旨在建设业界领先的面向营销员的、具有前瞻性和可扩展性的，符合主流技术的保险销售一体化平台，聚焦核心作业功能，体现支持、服务、提效和赋能。系统主要实现功能车险、非车险出单、业绩管理、客户管理、营销活动、商业计划书、续保管理等。通过两个视角挖掘，营销员视角，集获客、展业、服务、个人成长为一体，作业辅导始终伴随的创新工作模式、挖掘潜在销售机会，提高工作效率，促进职业能力发展；管理视角，综合管理招募、培训、业绩、活动、提供营销指导和线索及客户服务锦囊，降低消息传递成本、提升营销员团队整体产能和绩效。  
该项目于2019年10月正式启动，我担任系统架构师角色，负责系统总体架构设计工作。在系统的实现过程中，通过研究分析架构风格的各种特性，结合本项目实际情况，整体采用微服务风格，具体场景选择不同的架构风格。  
架构风格对于系统的分析和建设起着至关重要的作用，架构提供抽象框架的粗粒度模式，提供架构级的通用解决方案。这种设计重用可以极大提高系统建设进度并规避风险。常用的软件架构风格有数据流风格、调用/返回风格、独立构件风格、虚拟机风格、数据仓库风格。数据流风格包括批处理序列与管道—过滤器，每步处理独立，顺序执行，线性流程机制。调用/返回风格包括主程序/子程序风格，面向对象风格，层次结构风格，明确系统层次，降低系统耦合度。独立构件风格，进程通信，事件驱动，构件风格特性为重用提供基础服务。虚拟机风格包括解释器，基于规则的系统，自定义规则良好的扩展性及灵活性。数据仓库风格包括数据库系统风格，超文本系统风格，黑板系统风格，以数据为中心。除此之外还有微服务、SOA等架构风格。  
技术服务与业务，但怎样能更好的支撑业务的发展也是考虑的主要因素。综合衡量后采用微服务风格，紧密围绕业务能力开展去中心化。服务间通信交互采用服务架构SOA。每个微服务基于场景也采用不同的架构风格实现，如营销活动服务采用虚拟机解释器机制灵活定制规则。下面具体论述其应用过程。  
微服务架构模式，使用spring-cloud组件将单一应用程序按照业务领域划分单独的服务，每个服务运行在其独立的进程中，服务与服务采用轻量级的通信机制，每个服务围绕着具体业务进行构建，且能独立的部署。解决传统服务架构功能集中，发布部署运行在同一个进程中，应用复杂，臃肿存在大量重复代码，复用性较低，遇到性能瓶颈或架构重构时需花费大量成本投入。基于此按照业务领域拆分多个微服务架构解决上述问题，每个微服务粒度都很小专注具体业务功能，可单独部署发布。为进一步降低程序间依赖使用MVP分层，实现前后端分离，表示层注重UI交互用户体验，功能层连接表示层与数据层桥梁主要实现业务逻辑处理，数据层负责数据服务、缓存服务、文件服务的处理。微服务的灵活性、扩展性、维护性能够使我们的开发效率更高，沟通成本更低，响应速度更快，迭代周期更短，停机发布时不会影响其他微服务的正常使用。性能解决方案，采用Nginx反向代理服务机制，当访问量并发量高时还可以针对微服务增加服务器拓展。  
基于服务架构SOA风格，微服务内部交互使用Spring的Eureka组件搭建注册中心，服务提供方和消费方都在注册中心进行注册，服务交互时通过注册中心获取对方的服务地址信息，然后发起RPC远程调用。原则上提供的接口必须都是无状态、高可用、易于水平扩展节点的。多个服务实例之间完全对等，请求提交到任意服务器，处理结果都是完全一样的。其次，它涵盖了主要领域目前分为出单、业绩等部分功能的业务组装，实现对于一些复用性较强，与具体业务并无多大关系的内容，会尽量复用公共服务部分所提供的标准接口，尽量最大程度的实现基于现有业务的组合，而不必来一个业务，从头开发一个业务。系统与外部集成采用企业服务总线（ESB），外围系统交互涉及17个如核心、影像、销管、收付费、集中收款、ECIF、客户画像、增值服务、续保、理赔、标的库等。ESB提供了连接企业内部间现有系统的功能，服务注册命名管理，提供消息路由及寻址服务，支持多种数据格式转换，提供日志和监控功能。通过系统内服务注册中心，系统外的企业服务总线双重机制，集中服务的管理及监控，有效提高系统的透明性、高可用性。  
虚拟机风格，因项目整体按照微服务架构风格划分，每个微服务结合实际业务场景分析还会运用到不同的架构风格，例如营销活动服务，基于客户画像、埋点数据的分析，定制不同的激励方案拉力赛、基本法边缘状态提醒、车加人专项提升等各种活动，如果采用传统架构会基于不同的活动方案定制开发，使用的及时性不能保证，同时也消耗了大量的人力成本。针对该场景运用虚拟机的基于规则的风格，具备良好的灵活性，前端展示层由参数配置动态生成活动页面，规则采用ILOG，在营销员出单过程中，请求ILOG判断是否符合活动奖励条件，并获取ILOG返回的活动奖励结果。各种活动场景由业务在ILOG灵活配置，根据营销活动触发条件返回不同的奖励，降低了系统推广和维护的难度。  
整个项目历时9个月的实施，于2020年7月完成验收并顺利上线，日均出单保费规模达到千万级别，赢得了良好的用户口碑也在业界内树立了标杆。  
经实践证明采用微服务风格带来良好的扩展性、可用性、可修改性，但同时也会带来在分布式部署上繁琐的配置工作量，增加了部署失败导致的服务不可用，尽量减少人为导致的配置性错误，在系统运行稳定后采用自动化发布部署方案。同时微服务划分的粒度层次也是关键因素，公共基础服务停机将导致依赖的微服务也不可用，所以划分微服务要格外注意彼此的依赖关系，公共基础部分的服务尽量与具体业务并无多大关系的通用业务如（登录、权限）校验或常用的技术性较强的服务（如监控服务、批处理等），最大程度的复用弱业务性、强技术性的基础服务。目前还待改进的是系统对数据库的用户未按照微服务主题划分，后续将着重对数据库性能方面调整，拆分每个微服务单独部署数据库，基于ORM的mybatis技术封装数据库调整对应用架构影响在有限范围内，但对系统事务性能又将进一步提高。