电商项目微服务中台架构落地实战

图灵：楼兰

这一节课，我们开始进入电商项目的中台战略设计。中台大家应该都不陌生，现在到处都在提中台。但是中台是一个很虚的概念，我们这节课会来设计实现一个微型的中台，帮助大家理解中台。同时也是拓展大家的技术视野，给面试增加自信。

对于整个企业的业务运转来说，我们之前学到的所有技术、架构都是属于具体的务实的战术设计。而到了中台这一层，就开始进入整体的，务虚的战略设计了。所以对这节课的学习，重要的是要去理解做什么以及怎么做的问题，而不是怎么实现的问题。但是，这些务实的战术设计是构成战略设计的工具，所以，这一节课，如果同学们有些技术跟不上的，先不要纠结技术细节，先跟上整体的思路，回头再去补充技术细节。

一、大话中台

中台，是国内互联网中，继微服务之后又一个火热的概念。从2015年阿里提出"小前台+大中台"的概念后，中台成了国内微服务实施过程中绕不开的一个目标。但是，虽然中台的实施如火如荼，但是关于中台，依然是雾里看花，一百个人会有一百种理解。关于中台的知识，即使单独拉出一个专题课程出来，也不一定能说得清楚，所以这节课中不会去做过多讨论。这一讲关于中台的讲解只是作为一个载体，来分享一下我在参与了多个中台项目建设后的一些心得与体会。

在这节课中，我们会以一个支付风控系统为例来讲解中台的概念。但是，同学们需要理解，其实这种方式是不太合适的。我们课程中这个风控系统甚至包括整个电商项目，其业务体量是不足以支撑中台这么大的一个概念的。或许课后你能意识到，这个电商项目，包括我们这节课要搭建的风控系统就可以是一个中台，但是他是整个中台战略中的一个部分。所以，在这节课中，你需要能够站在更高的角度来看后续的技术内容。这个高度要比程序员高很多，要站在架构师、总经理甚至总裁的角度来理解中台。因为关于企业中台的建设，如果不是从一个与业务平等甚至比业务更高的管理层面来推动，基本上是行不通的。强行建设的中台，必然退化成一个业务的附属品。另外，**你也需要有一定的业务想象力，才能理解中台的建设思路，而不是像之前的课程那样，关注于各个技术实现的细节。**

前言交代完了，本节课会从以下几个角度来讲解中台的概念。

什么是中台?

中台这个概念，最早是由阿里在2015年提出的"小前台，大中台"战略中衍生出来的一个概念，他的灵感来自于一家芬兰的小公司supercell。这是一家号称是全世界最成功的移动游戏公司，说这个公司你可能没什么印象，但是说到他开发出来的游戏，你可能就会有点印象了。supercell可以说是全世界最会赚钱的游戏公司，旗下推出的每一个游戏都非常火爆，包括《部落冲突》《皇室战争》，还有《卡通农场》、《海岛奇兵》、《荒野乱斗》。他有个显著的特点，就是在公司内部，任何人都可以提出自己的游戏点子，然后在公司内组织自己的项目团队，称为cell。每个cell人数不多，一般不会超过7个。每个Cell内自行组织游戏的设计、开发、快速推广。然后以最快的时间推出产品的公测版，看看游戏是否受到用户的欢迎。火爆的游戏就存活下来，反响平平的游戏就快速终止。期间几乎没有管理角色的接入，同时失败后，团队也不会收到惩罚，团队成员总结项目经验后，迅速转入其他Cell。而阿里正是在对supercell进行考察之后，才有了中台这个概念。于2015年正式开始推行中台战略，并指定了为期三年的中台战略考察期，并迅速成为了互联网企业的标杆。

所谓中台，就是将各个业务线中可以复用的一些功能抽取出来，剥离个性、提取共性，形成一些可复用的组件。通过这些组件，就可以使日后的系统开发成本降低，质量提高。**大体上，中台可以分为三类，业务中台、数据中台和技术中台。**

- 业务中台就是抽象出来，在各个业务线都可以共用的一些业务组件，像用户权限、会员管理、移动支付等等这些公用组件，可以在各个业务线中都使用。
- 数据中台是对整个业务系统的数据进行统一存储、建模与计算，为各个业务系统的数据分析与利用提供支持。
- 技术中台就是要封装各个业务系统所要采用的技术框架，使上层业务开发的门槛降低，提升交付速度。

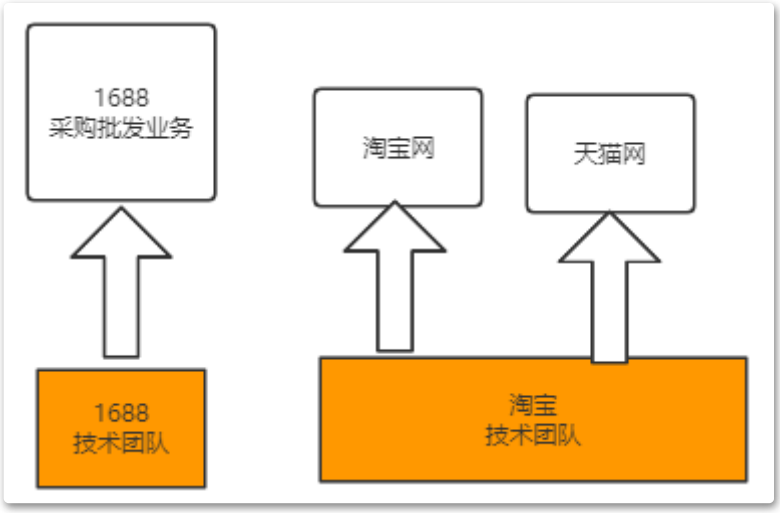
> 可以设想一下，如果你是一个架构师，你需要将全公司的软件资源进行整合，你会怎么考虑？是整合数据还是整合服务？

另外对于中台，你需要有一个清醒的认识，虽然这个东西现在很火，互联网开口闭口都离不开中台。但是中台只是一种战略设计思想，他也有他的局限性，并不是适用于所有的软件场景。另外，中台也并不一定就是好的。中台战略涉及到对整个公司的资源进行大刀阔斧的调整，从业界的情况来看，成功的案例固然很多，但是失败的案例也不在少数。网上可以查到很多中台失败的案例，甚至在阿里内部，也不断传出要拆中台，唱衰中台的声音。所以，中台的好和坏，需要每个人自己有自己的评价与思考。

> 网上的讨论文章很多，推荐一篇给大家看看 https://blog.csdn.net/yellowzf3/article/details/107551908

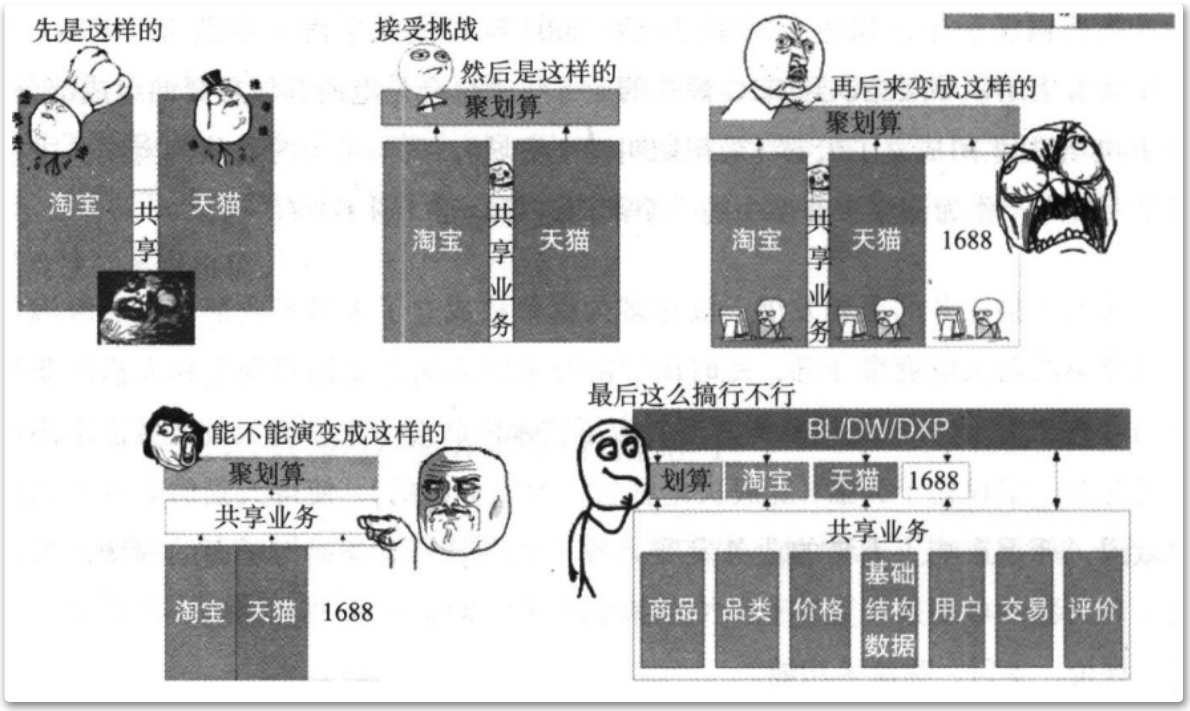
怎么建设中台?

中台的建设，也不仅仅是技术方面的问题，而是涉及到整个企业内资源调度的综合性问题。很多企业在组件软件开发团队时，都会优先基于业务的横向分割，形成产品、设计、开发、测试、运维这样大的组织结构，称为烟囱式团队。例如阿里早期的团队建设是这样的：



这种模式没有太多的历史技术和业务的包袱，在业务简单、参与的人比较少的时候，往往可以获得一个比较快的支付速度。但是随着项目的不端推进，业务变得越来越复杂，参与的团队会越来越多，这时候这种烟囱式的团队建设就会体现出很多的弊端。例如最为明显的就是功能重复建设以及维护会带来重复的投资。另外更深层次的问题在于，这种烟囱式的团队，团队之间的沟通和协作会非常困难。即不利于企业的业务经验沉淀，也不利于技术梯队的建设。在企业长期的发展过程中，就会体现出人力资源内耗严重，项目新需求研发越来越慢，交付的产品质量越来越低。

例如阿里在09年就成立了共享事业部，尝试对于烟囱式团队进行改造。经过多年的摸索，才逐渐将平台化建设的思路给梳理清楚，并形成了中台这样一个具体的体系。

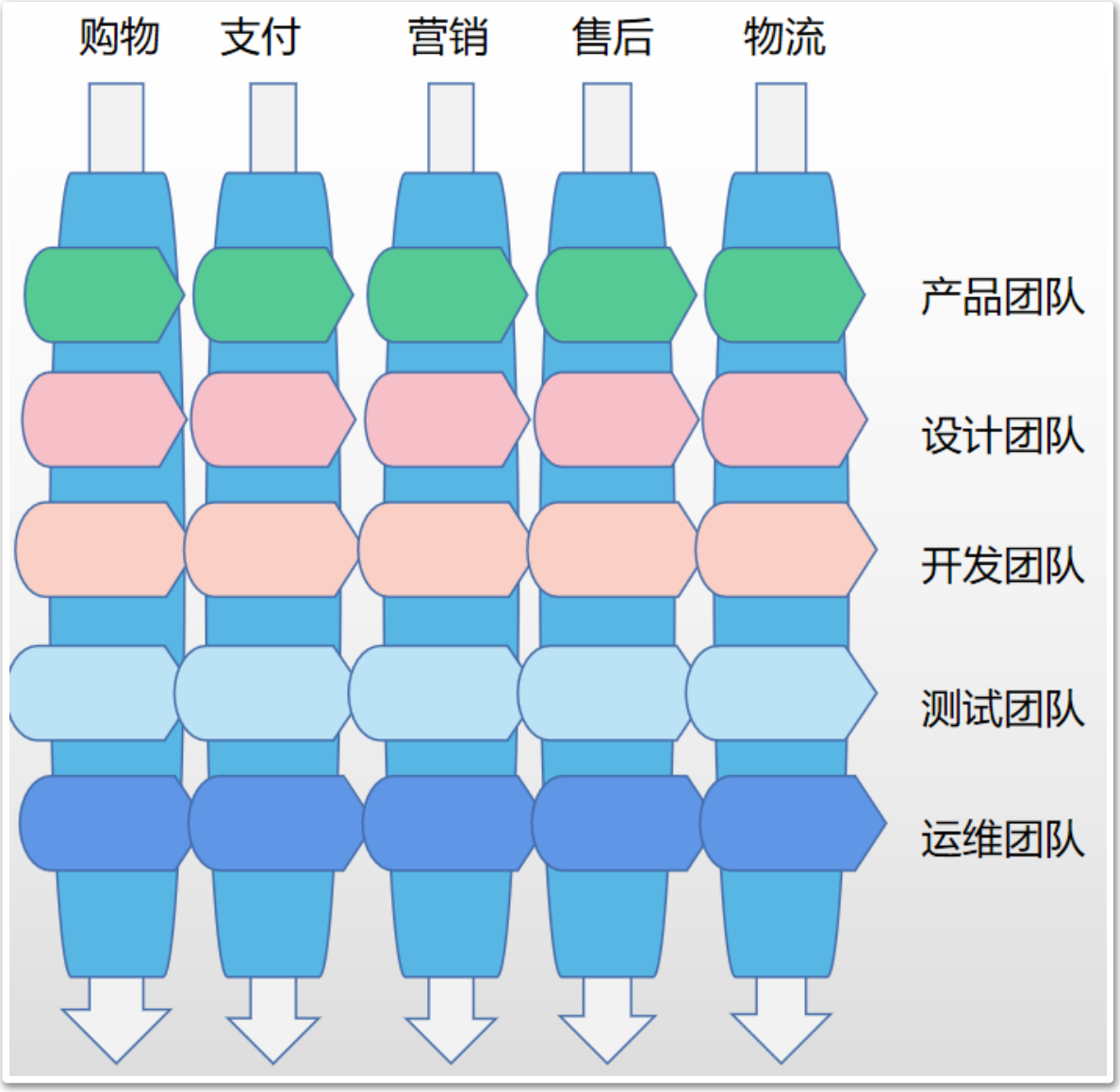


究其原因，是因为从需求到交付的整个过程中，要经历的部门越来越多，部门之间的沟通成本也越来越大。例如要开发一个完整的需求，需要找各个团队要人，而分配过来的人可能又不懂这一方面的业务，互相沟通就会消耗大量的资源。于是也有企业会按照业务线划分业务部门，甚至有的大型项目，也会以业务为单位，外包给不同的公司，组建跨多个公司的复杂团队。但是随着业务部门的划分，业务之间的沟通又变得更加艰难。在面对复杂业务时，无法及时做出变化，造成团队响应缓慢，跟不上业务变化，从而在互联网的竞争当中错失先机。

而具体到我们所讨论的代码层面，烟囱式的团队必然造成烟囱式的开发。每个团队都按照自己的技术栈形成自己的代码库。这里面有大量的像JDBC连接池这样的重复的功能。代码量增多了，其中隐含的BUG也会增多。测试团队所要覆盖的用例也随之增多，运维团队需要保证的维护方案也跟着增多。这样，整个团队的工作量一直增加的同时，代码的交付速度与交付质量也就跟着下降了。

如何打破这个问题呢？阿里的中台战略就提出需要打破部门之间的隔阂，将横向的烟囱式架构竖过来，以业务需求为单位形成一个个快速组合的**需求团队**。每个团队都直接面对客户。例如电商中，要开发购物车功能，就由产品经理牵头，组建购物车团队，负责整个功能的设计、开发与上线。购物车功能开发完后，团队就解散，去做其他的功能。而当购物车功能需要进行变更时，再重新组建起需求团队，团队成员对整个购物车功能都是相当熟悉的，团队之间的沟通也是非常顺畅的。并且，由于团队人员不断的经历不同的业务线，企业内部的部门信息墙也就自然的被打破了。

然后再结合微服务的技术架构，每个团队可以自己维护自己的微服务。开发完成后，也可以自行打包，独立发布，不需要等待其他团队，这样交付速度自然就提升了。



然而，组建这样的团队对成员的要求是非常高的。每个需求团队中都需要有一个或多个组织者，这些组织者必须即懂业务，又懂开发，甚至还要懂测试和运维。所以在阿里内部，诞生出了非常多的业务架构师，这些业务架构师不同于一般的架构师，即能讨论技术架构，又能设计业务框架，无论从哪个方面来说，都是牛人。这种模式虽然好，但是如果每个企业，每个团队都照这个标准来组建，那企业的人力成本就会过高。这样的团队建设是没办法真正落地的。那如何解决这个问题呢？**解决问题的关键就在于中台。**

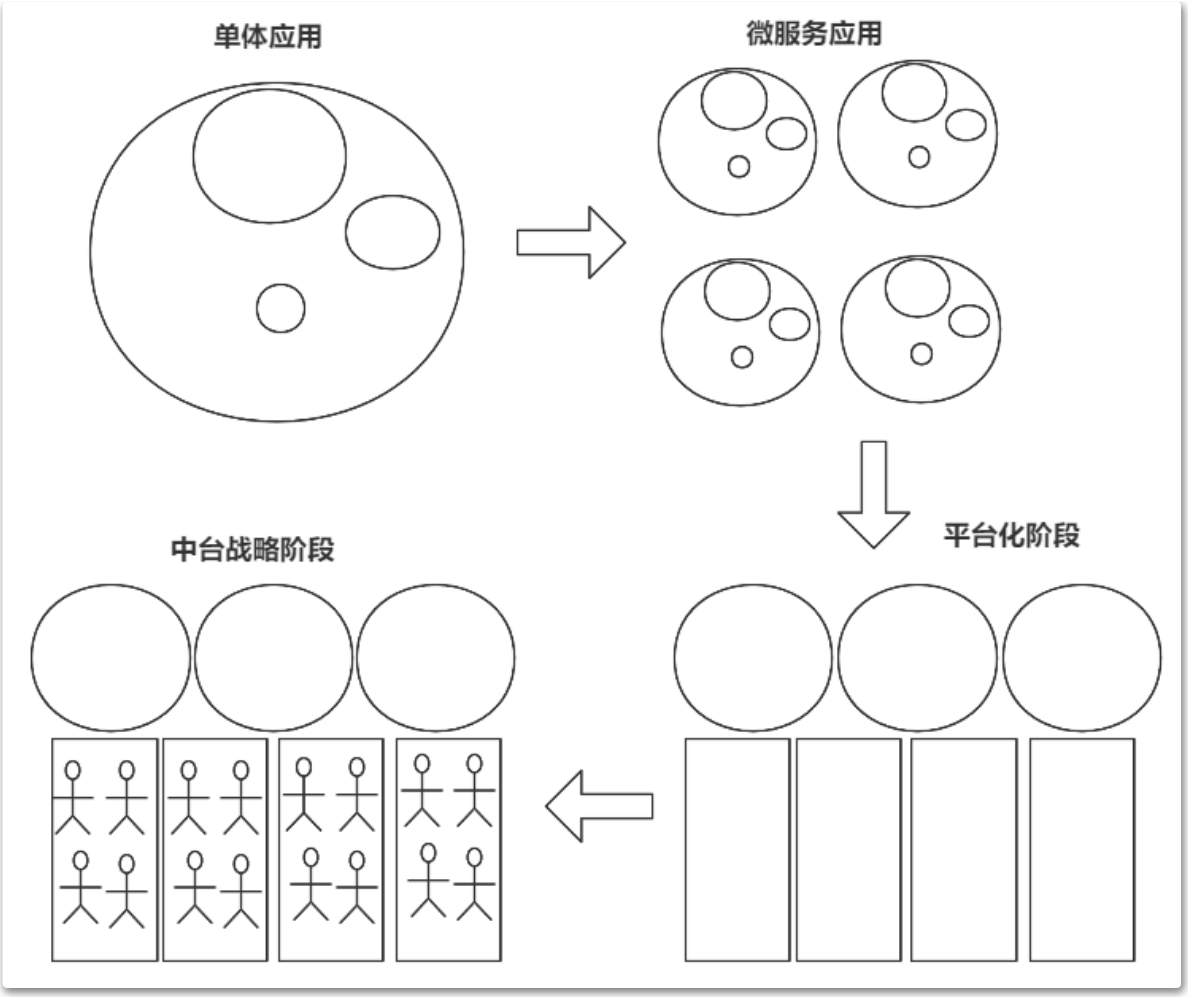
中台的本质就是提炼各个业务的共同需求，进行业务和系统抽象，进而形成通用可复用的业务模型，打造成组件化产品，形成系统化的能力输出，供各个业务部门使用。简单来说，就是业务需要什么业务，需要什么资源，可以直接找中台，不需要每次去改动自己的底层。例如，如果一个支付企业在电商、零售等行业已经形成

了非常丰富的支付业务能力，支持了非常多的支付场景，就可以沉淀形成支付中台。后续如果需要开发团购业务，需求团队就完全不需要自己开发支付方面的功能模块，直接找中台要就行。而在团购业务发展过程当中，有可能出现一些新的业务场景，需求团队可以交由支付中台统一提供支持，这样支付中台又能够进一步扩展自己支持的支付场景，以后在设计新的业务时，提供更全面的支付场景支持。这样，新业务的设计与开发可以像拼积木一样简单快速，而中台也能在业务实施过程中，不断吸收养分，逐渐壮大起来。整个中台体系在企业内就形成了一个良性循环。

在中台建设过程中，数据中台、业务中台和技术中台往往是一个相辅相成的建设过程，针对不同类型的中台也会有不同的建设重点与思路。下面我们重点讨论下技术中台的建设方式。

怎么设计一个可落地的技术中台?

中台战略的形成，往往需要经过三个阶段的转化。**第一个阶段是从单体应用到微服务应用的转化**：这个阶段跟业务系统转化是相同的。**第二个阶段是从微服务应用向平台化的转化**：在这个阶段会逐渐将各个业务线的共有能力提炼并沉淀，集中形成一个一个的平台。**第三个阶段是从平台化建设向中台战略的转型**：经过平台化的建设阶段后，技术中台就基本成型了。在以后的阶段，针对这些技术平台，再逐渐调整公司的整个组织架构以及资源投入，就形成了完整的中台战略。



这个技术中台是更为广义的技术中台。在这个技术中台中需要实现的，是统一各个需求的技术栈以及由此衍生出来的测试、发布、运维等一系列的技术动作。通过这样的技术中台，各个业务团队的技术门槛得到极大的降低，开发的工作量减少了，测试运维的工作也都比较固定了，整个团队的工作模式也就越来越高效，可以将更多的精力集中用来深刻理解业务，并快速响应业务需求的变化。

而**在中台建设过程当中，技术中台往往是最为重要的一环。**技术中台团队，往往需要由最精锐的开发人员组成。这样的技术中台需要功能全面、简单易用、部署简单、易于升级等等非常多的特点。下面列出几点我之前参与建设的技术中台的特点，或许能够让你对技术中台有个大致的概念。

- 技术栈统一：采用SpringCloud技术栈，提供统一的架构支持。同步服务调用、异步消息通信，再到数据存储等功能，提供基于SpringBoot的拔插式支持，各应用只需要按需组合即可。在此基础上，提供统一配置、统一版本管理等集中式的管理方案。

> 例如框架中各种maven依赖的版本，只提供几个统一的框架版本供业务进行选择。框架版本内的各个组件版本采用集中配置的方式也集中到maven仓库中进行管理。这样，框架版本内部对某些组件版本进行版本微调时，对业务可以做到基本无感知。

> 或许你听说过在2020年fastjson爆出了一个高危漏洞，在行业内造成了非常大的影响，各个应用都需要尽快升级fastjson版本。而在我们的技术中台架构中，只需要在maven仓库中对fastjson的依赖版本进行统一升级，各个业务应用基本上不需要做改动，只需要随着迭代进行一次常规升级就避免了这个问题。

- 解决方案统一：对微服务调用、MQ异步通知、日志脱敏、传输数据加密、缓存一致性等各种功能在框架中提供统一的解决方案。这些解决方案，大部分是以API的形式提供，业务方只需要进行调用即可。而如果不能形成API接口的，集成到代码静态检查规范当中，在编译发布的阶段给出统一的指导。
- 运维与框架统一：将外部依赖的组件与框架形成统一。例如某业务可能需要用到MQ做异步通信，会由技术中台团队完成MQ的部署，并且将MQ相关的实现以及配置信息一并上传到配置管理中心。业务团队在使用时甚至都不需要知道MQ部署地址，就可以直接拿来用。
- 部署与运维统一：以Jenkins为基础，定制整套完整的部署运维方案。业务团队只需要往Git仓库中提交代码，后续项目打包以及测试环境的部署全部自动完成，不需要人工参与。
- 上线方案统一：对线上环境，机器配置与服务部署都形成统一的标准，业务团队申请线上资源时，只需要申请自己需要什么，而不用管具体的细节。

所以，技术中台并不等同于框架设计，他的落地方案是多种多样的。我们可以说用SpringCloud做微服务是最好的，但是对于中台，很难说某一个公司的落地方案就一定是好的。不要简单的认为跟着互联网大厂学习了一下他们的技术或者架构，我们就做好了中台了。中台没有最好的，只有最适合的。从这个角度来说，我们这个简单的风控中台，对于我们这个电商项目，就可以是最好的中台。

二、搭建风控平台

1、业务要求

到目前为止，我们的电商项目业务功能算是比较成体系了，但是对于风险控制方面，还没有太多考虑。现在有个需求，要在注册、下单、交易等各个环节来增加对用户手机号码安全性的检查。你会怎么做?

最开始的做法肯定是在各个业务环节里加代码，去检查手机号码是否安全。然后有些逻辑很相似，就会抽象成一个微服务。当微服务的逻辑逐渐复杂，使用场景开始增多，就会独立出一个平台。**这个平台要求有丰富的业务能力，并且能够给多个外部系统调用。**现在要你来设计这样一个平台，你会怎样设计？

> 大概讨论几分钟，引出一些网上的服务接入平台的接入模式

然后来了解下我们设计出的这个风控服务平台。

代码参见GenSI2项目。分三个方面来讲解项目。

2、整体业务流程设计

服务端：

分为三个统一的业务接口，一个swagger的测试接口。 另外两个是需要加密的客户端接口，一个是同步服务调用接口，另一个是异步服务通知接口。三个接口的业务接入逻辑略有不同，但是后面的业务实现逻辑是一样的。

业务实现通过serviceCode来区分。每个serviceCode标识一种业务能力，匹配不同的业务报文。

提供同步服务调用和异步服务通知两种接入方案。

客户端：

先接入，再使用。需要使用接入后分配的相关信息，才能调用具体的服务。

2、架构设计

服务端拥有完整的业务流程，单独记录每次服务的日志。对外提供基于HTTP的通用业务服务，对内通过Dubbo调用内部的服务能力。

3、代码实现的关键点

请求报文加解密

重复请求处理

日志文件处理

大家可以想象一下，我们这个风控平台是不是已经具备了往中台发展的潜质？以后完全可以通过ServiceCode，扩展出更多更丰富的业务功能，给电商项目提供更多的支持，甚至给电商以外的其他系统也能提供支持。但是，即便他的能力很强，也只能称之为平台，离中台还有一定的距离。下面，我们来理解下中台。

三、课后作业：与电商项目订单系统集成

GenSI中搭建了一个简单的客户端，所以与商城订单的集成就不再去演示了。

基本步骤：

1、订单系统需要引入客户端包 ftClient.jar。

我们的开发过程是将ftClient.jar以本地文件的形式直接引用到项目的maven依赖中，这是一种不推荐的开发方式。如果是公司内开发，可以将ftClient.jar上传到公司的maven仓库，然后从maven仓库中引用。

2、在GenSI中先注册应用。然后在订单系统 tulingmall-order-sharding 中集成客户端调用工具。

3、在下单时增加读取手机号码的标注，如果包含诈骗这一类的关键字，就拒绝下单。

四、怎么将我们的风控平台建设成为中台

中台是一种企业战略，而不是一项具体的技术。中台战略涉及到的不光是技术重组，还包括公司的组织架构、人员流动机制、资源投入等各个方面的协调。所以如果你站在高层视角去看过中台，那么理解中台需要发挥你的想象力。

这里通过几个问题的思考，带大家站在架构师的角度，从大到小形成一个可实施的中台建设方案的设想。

1、中台体系问题：

问题1：你可能觉得我们这个风控中台业务太小了，那一个企业级的风控中台是怎么发展出来的?

1》从小了说，以我们的这个电商项目的风控部分需求为例。只是我们这个项目业务体量还没有上来，所以你可能还没有太多的感觉。

一方面风控需求的面会越来越广。

首先，目前我们的这个中台，还只有两个业务，业务体量确实比较少。但是当以后业务多了之后，可以通过ServiceCode快速扩展新的业务能力。

然后，目前中台对接的业务系统还只有电商这一个系统。随着业务体量的加大，用户、营销、交易都可能发展成为独立的业务系统，这个时候风控中台的体量就大了，还需要针对客户端进行接口权限控制、调用计费控制、数据权限控制等等。

接下来，针对后台数据，目前还只是简单的采用搜索引擎的数据，但是业务体量大了之后，必须要形成自己的数据体系，获取数据以及处理数据的能力都需要大幅度提升。例如对于用户，有大量追逐小利的羊毛党，只要稍不留神，一个营销活动就可能造成平台大量亏损。还要防止商家与用户勾结，进行线上洗钱等非法活动(虚构产品，交易不发货)。另外还有大量的国家监管体系也需要接入。这些行业内的运营经验，需要有一个能够集中进行整理以及沉淀的地方。而对于这些行业经验，全部由自身业务推动，积累得就太慢了。所以，这些经验和能力，单独进行沉淀，就逐渐形成了风控中台。

这些功能都考虑进去，是不是有点做中台的感觉了?

另一方面各个业务的风控能力也是不一样。

例如我们的这个电商项目，客户端应用的接入，不是一个简单的注册就能解决的，还需要有线上线下各种综合的审批制度。而如果上线运营后，单靠自己一家的数据，各自为战，是很难做到有效控制的。例如羊毛党，可以在平台之间迅速转移，像蝗虫一样导出收割福利。这也需要风控中台与行业内形成统一联防机制。然后针对营销环节，想要提高营销活动的用户转化率，也是需要风控中台提供一些数据支撑的。

而这些事情，都不是仅仅依靠技术就能够解决的，还需要有大量的人力、物力等资源的投入才能完成的。要将这些资源整合到一起，是不是有点做中台的感觉了?

有一句很经典的话：中台，是业务演进出来的，而不是架构出来的。实际上，阿里早在09年就开始建设共享事业部，当时阿里还只有淘宝和天猫两个产品。但是从"厚平台，薄应用"的平台化战略走到中台战略，阿里积累了近十年的建设经验。

2》从大了说，当中台体系足够完善，就可以用成熟业务场景反哺相对不太成熟的业务场景。

例如阿里的无人超市、无人酒店、盒马生鲜等新业务，开展起来就非常快。因为风控、支付、用户、商户这些环节都已经有了现成的能力，线上部分只需要开发一个简单的APP，将这些能力组合到一起，就可以开展业务。

问题2： 有了风控中台的概念后，你觉得我们电商项目还有哪些地方可以抽取成中台?

从技术体系上，风控中台肯定不是单独搭建的，还需要有用户中台、商户中台、收银中台、交易中台、营销中台等等其他中台一起发展。

问题3： 现在人人都在喊中台，是不是所有企业都要用中台?

虽然现在看来，中台确实是非常好，但是业界尤其是阿里、京东这些中台战略领先实施的大厂，也有不少的声音开始反对中台了。为什么呢？这是因为中台的理想虽然好，但是落地实践的问题却非常多。所以如果你所在的公司准备要开始推行中台化了，有些问题就需要提前考虑好了。

1、中台的价值很模糊。

中台并不能直接产生价值，他需要依托于业务部门才能体现出他的价值。而一个企业的业务往往是不平衡的，有赚钱的核心主业务，也有不赚钱的一些边缘小业务。这个时候就很容易陷入一种中台抱大业务大腿，而小业务抱中台的大腿的困局。中台往往优先按照大业务的要求进行功能开发，而小业务的要求就比较难得到相应。这就很容易造成新业务的同质化，从而影响到新业务的竞争能力。其实这从阿里也能看出点端倪，阿里的无人售货机、无人超市、无人酒店这一类的业务，其实就线上能力这一块来说，都是很相似的。如果阿里不是不断的通过拓展线下环境，那线上的很多产品就都沦为差不多的东西了，大家用起了也会丧失新鲜感。阿里的体量尚且如此，一般的企业就更需要慎重了。

2、中台并不是总能提炼出共性需求

中台的核心是提炼业务的共性，进行沉淀、提炼，形成中台业务能力。但是在具体实施时，哪些功能是通用的，哪些功能是个性化的，很难形成一个标准。这个度很难把握，中台做得太多了，就成了业务系统偷懒的利器，反正什么功能都提给中台就行了。中台做得太少了，又很难得到业务部门尤其是小业务的反馈，评估不出真正通用的功能。中台很容易变成大业务的后台，而对小业务，一句"你的功能不通用"，就全部给打回去了。

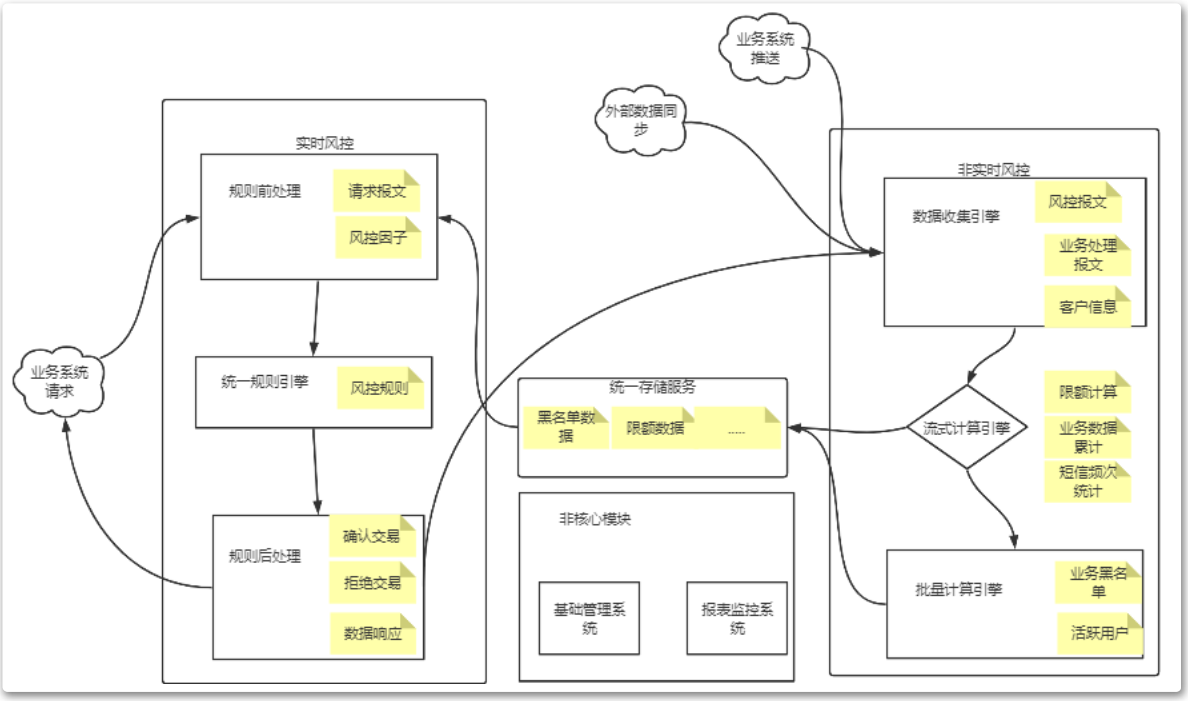
3、中台存在更多的变化

中台的业务能力要实时发生变化，这很容易造成具体业务掉队。例如一个中台针对十个业务开发了一个底层业务能力，这时，其中一个大业务要对这个业务能力进行更新。中台如果接受了，就很可能会要求其他九个业务也要调整自己的业务流程以适应新的中台功能。那这对这九个业务来说，就是无意义的工作。

所以现在越来越多的人提出中台并不是银弹，并不是所有企业都需要中台。

2、中台架构问题：

我们这个中台还太过简单，你可能会感觉困惑，这就是一个中台了吗？ 这是因为我们这里只考虑了数据的使用，而忽略了数据收集与处理的问题。一个典型的企业级风控系统的整体架构设计图。



3、中台技术问题：

在我们这个风控中台体系中如何快速开发一个新的业务？有没有发现要开发一个新的功能，要改的地方还是挺多的，开发过程也挺麻烦的。并且，所有的ServiceCode都在BusiService中通过if-else语句来扩展，可以想象要开发十个二十个业务后，整个代码会混乱成一个什么样的程度。

要如何保证系统优雅、稳定的演进？下一节的DDD就是现在业界非常认可的一种思路。