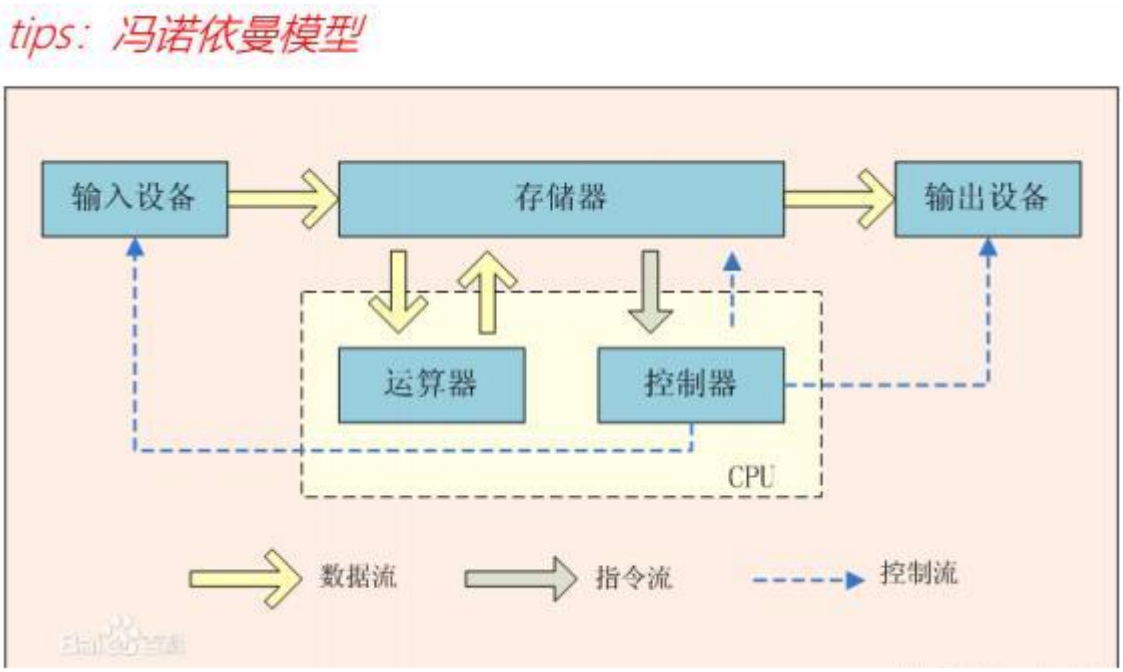
系统架构演进

一、基础知识

1.1 计算机

计算机的组成是有五部分完成的,分别是:输入设备,输出设备,存储器,存储器里面由运算器和控制器,



阿里巴巴在2009年发起了一项去“IOE”的驱动

IOE指的是IBM的小型机,Oracle的数据库和EMC的高端存储设备，2009年的去IOE的运动,一直到2003的支付宝的最后一台IBM的小型机的下线

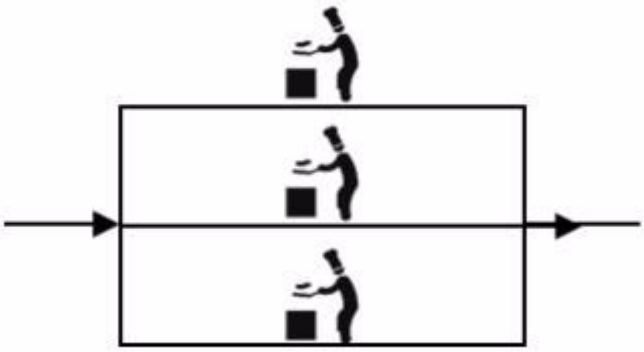
为什么要去IOE?

阿里巴巴过去数据库使用的是Oracle,并使用小型机和高端存储设备提供高性能的数据处理和存储服务。随着公司的业务量的上升,用户规模的不断上涨,传统的集中式的架构Oracle数据库在扩展方面遭遇了瓶颈。向传统的Oracle,DB2都是以集中式的为主,存在的缺点就是扩展性的不足,集中式的扩展主要是采用的是向上的扩展不是水平的扩展,这样时间长了，早晚都会遇到系统瓶颈。

1.2 集群

多台机器连接在一起共同工作，组成的就是集群

小饭店原来是一个厨师，切菜洗菜备料炒菜全干。后来客人多了，厨房一个厨师忙不过来，又请了个厨师，两个厨师都能炒一样的菜，这两个厨师的关系就是集群。



1.3分布式

为了让厨师专心炒菜,把菜做到极致,又请了个配菜师负责切菜,备菜,备料,厨师和配菜师的关系就是分布式的,一个配菜师也忙不过来,有请了个配菜师,这两个配菜师的关系就是集群了。所以说有分布式的架构中可能有集群,但集群不等于有分布式。

1.4 副本机制

副本指的是在分布式系统中为数据或服务提供冗余。

数据副本指在不同的节点上持久化同一份数据,当出现某一个节点的数据丢失时,可以从副本读取数据。数据副本是分布式系统中结果数据丢失的唯一手段。

服务副本表示的是多个节点提供相同的服务,通过主从关系来实现服务的高可用方案。

二、项目架构演进

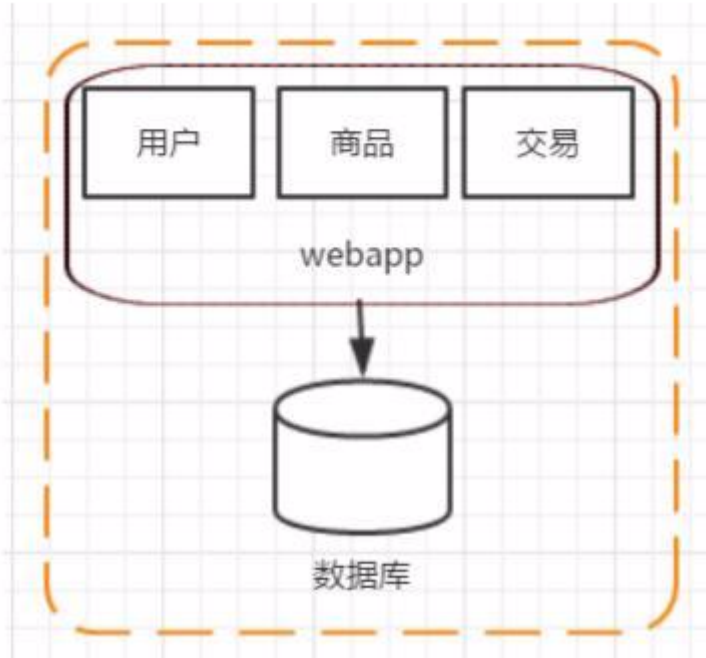
2.1、LAMP项目

单应用架构，小型系统应用程序、数据库、文件等所有的资源都在一台服务器上通俗称为LAMP

特征：应用程序、数据库、文件等所有的资源都在一台服务器上。

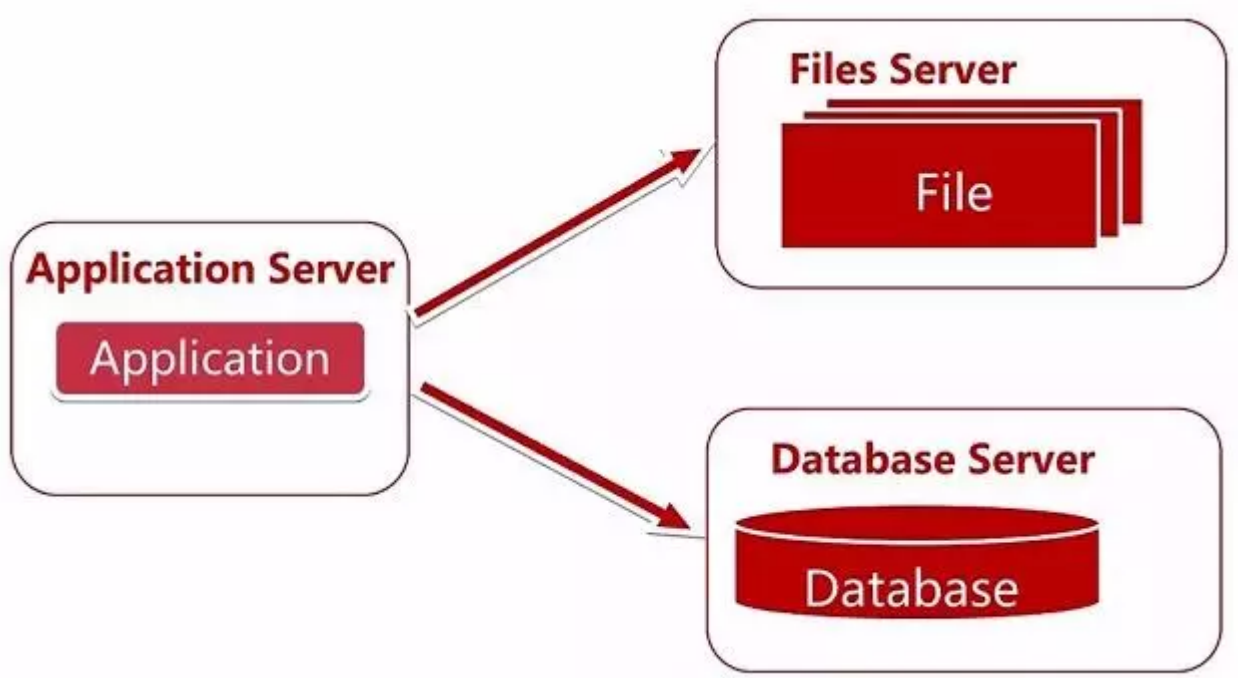
描述：通常服务器操作系统使用linux，应用程序使用PHP、Java、C#等开发，然后部署在Apache、Tomcat上，数据库使用Mysql，汇集各种免费开源软件以及一台廉价服务器就可以开始系统的发展之路了。

`all in one` 架构



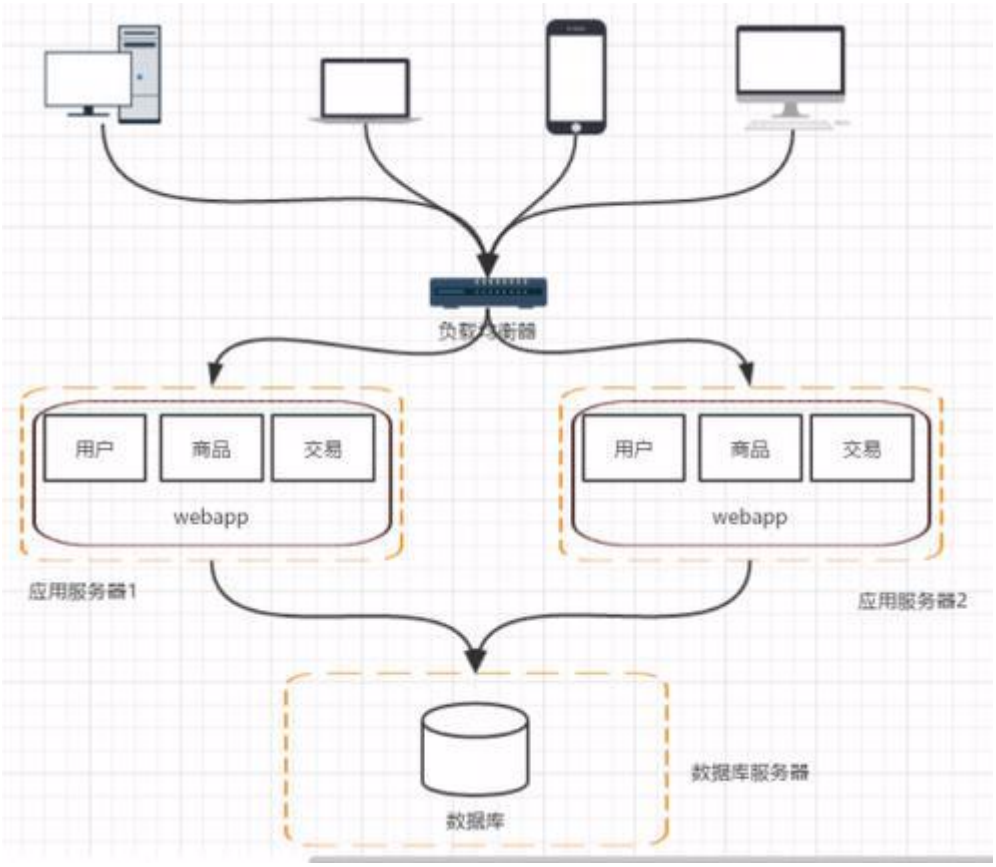
2.2、多机部署

随着网站用户逐渐增多，访问量越来越大，硬盘、cpu、内存等开始吃紧，一台服务器难以支撑。看一下演进过程，我们将数据服务和应用服务进行分离，给应用服务器配置更好的cpu、内存等等，而给数据服务器配置更好、更快的大的硬盘，如图所示用了三台服务器进行部署，能提高一定的性能和可用性。



2.3、服务器集群

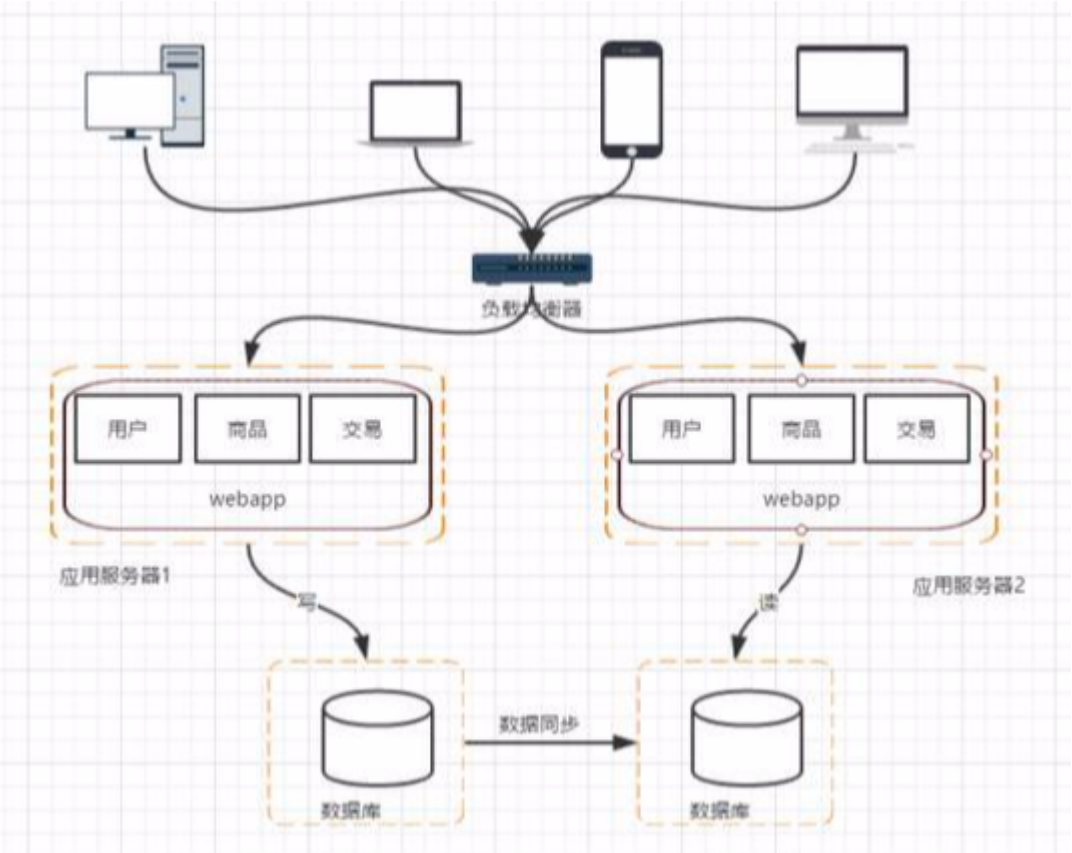
随着访问量和流量的增加,假设数据库没有遇到瓶颈,对应用服务器集群来对请求进行分流,提高程序的性能。存在的问题:用户的请求由谁来转发,session如何来管理的问题



2.4、数据库读写分离

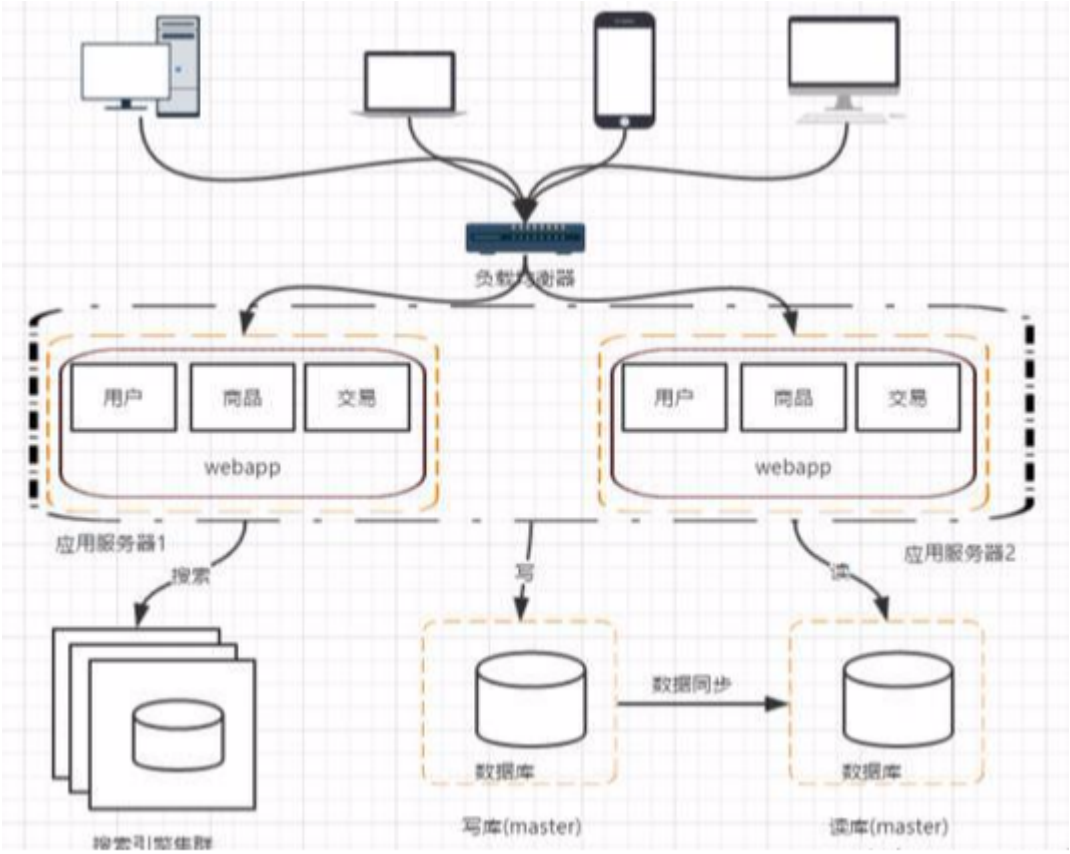
数据库的读与写操作都需要经过数据库，当用户量达到一定量时，数据库性能又成为了一个瓶颈，

可以使用数据库的读写分离，同时应用要接入多数据源。通过统一的数据访问模型进行访问。数据库的读写分离是将所有的写操作引入到主库中`(master)`，将读操作引入到从库中`(slave)`，此时应用程序也要做出相应的变化，我们实现了一个数据访问模块`(data access module)`，使上层写代码的人不知道读写分离的存在，这样多数据源的读写对业务代码就没有侵入，这就是代码层面的演变。



2.5、搜索引擎

数据库做读库的话,常常对模糊查询的性能不是很好,特别是对于大型的互联网公司来说,想搜索的模块就比较核心了,这是可以使用搜索引擎了,虽然可以大幅度的提高查询的速度,但是同时也会带来一些问题比如索引的构建。

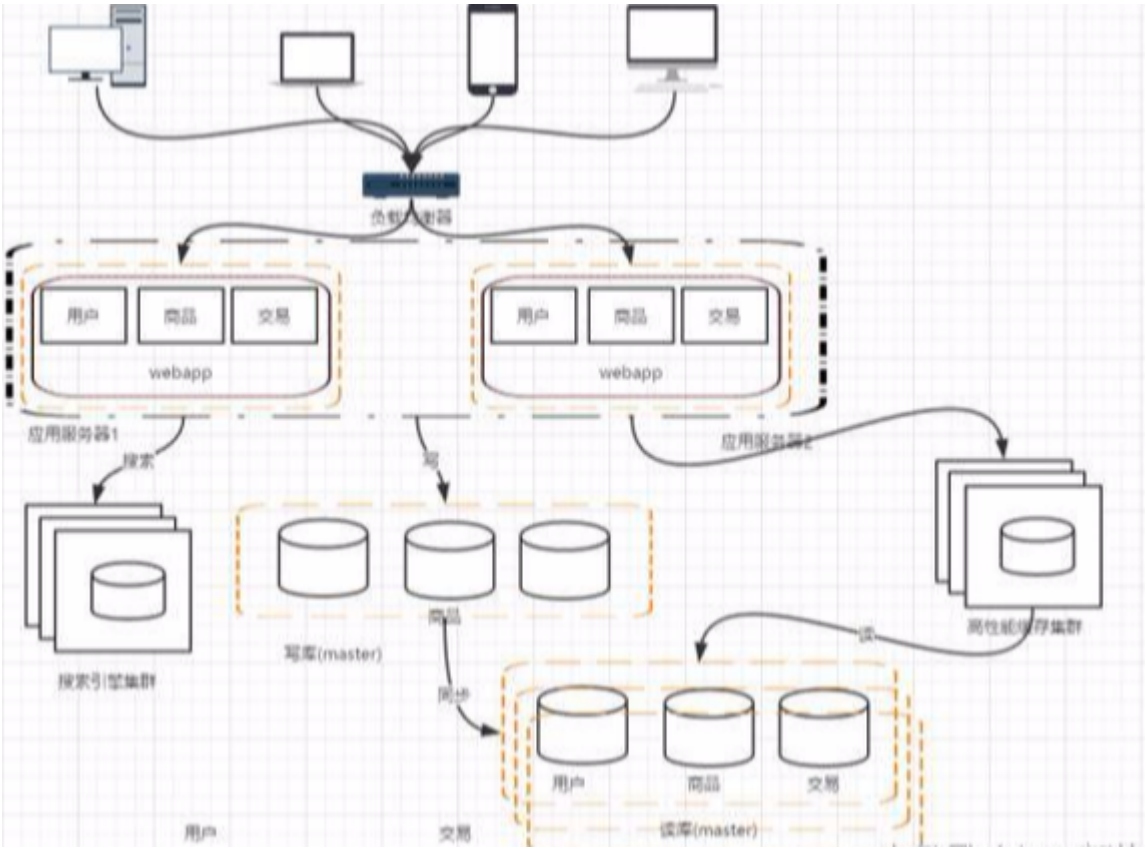


2.6、缓存

随着访问的并发越来越高，为了降低接口的访问时间提高服务性能，继续对架构进行演进。

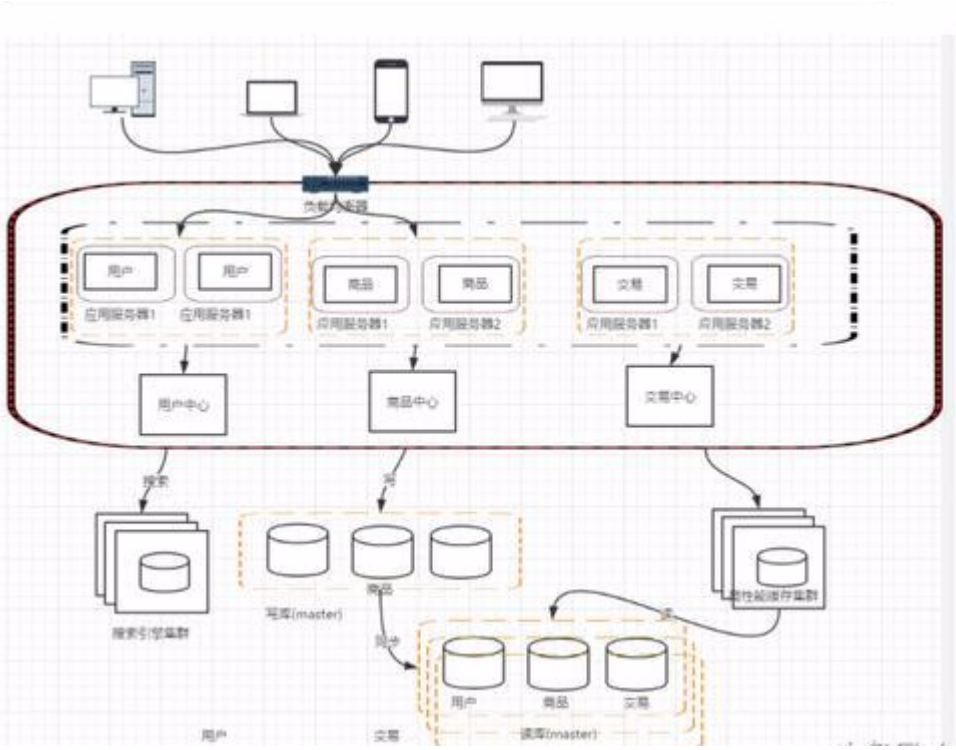
我们发现有很多业务数据不需要每次都从数据库中获取，于是我们使用了缓存，因为80%的业务访问都集中在20%的数据上(二八原则)，如果能将这部分数据缓存下来，性能就能提高很多，缓存又分两种，一种是Application中的本地缓存，还有远程缓存，远程缓存又分为远程的单机式缓存和分布式缓存。

对一些热点的数据,可以使用redis，memcache来作为应用层的缓存;另外在某些场景下,可以使用mongodb来替代关系型数据库来存储。



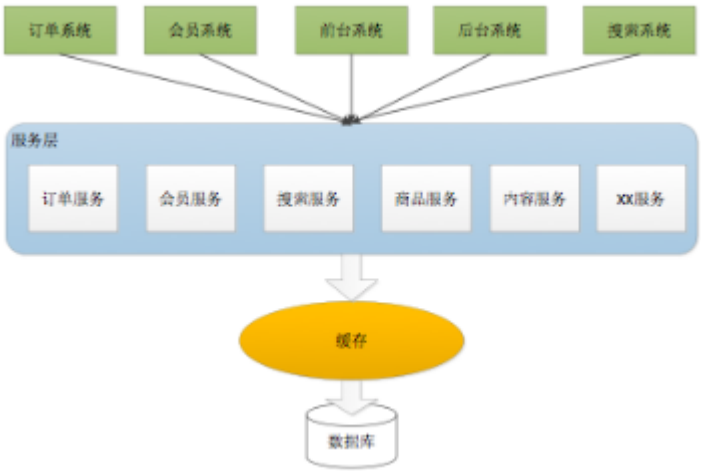
2.7、功能模块拆解

随着业务的发展,业务越来越多,应用的压力越来越大。工程规模也越来越庞大。这个时候就可以考虑将应用拆分,按照领域模型将我们的用户，商品，交易分拆成子系统。这样拆分以后,可能会有一些相同的代码,比如用户操作,商品的交易查询,所有会导致每个系统都会有用户查询和访问相关的操作。这些相同的代码和模块一定要抽象出来。这样有利于维护和管理。



2.8、SOA架构

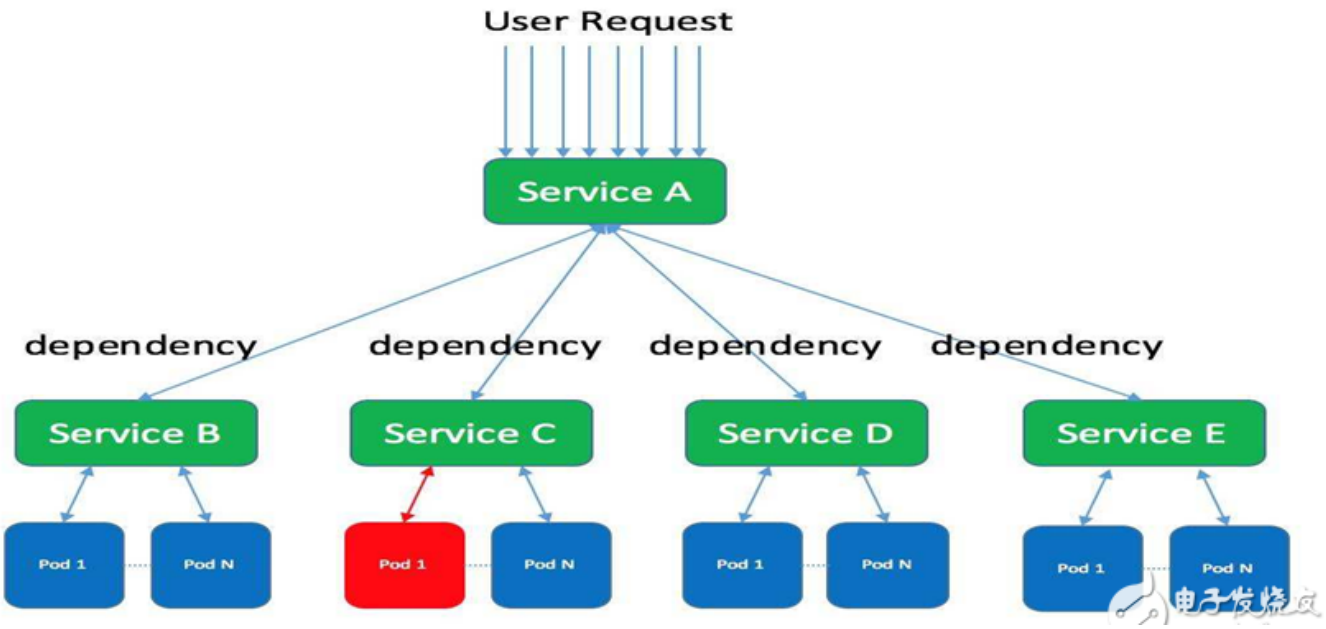
SOA:面向服务的架构（Service-Oriented Architecture）,也就是把工程都拆分成服务层工程、表现层工程。服务层中包含业务逻辑，只需要对外提供服务即可。表现层只需要处理和页面的交互，业务逻辑都是调用服务层的服务来实现。工程都可以独立部署。

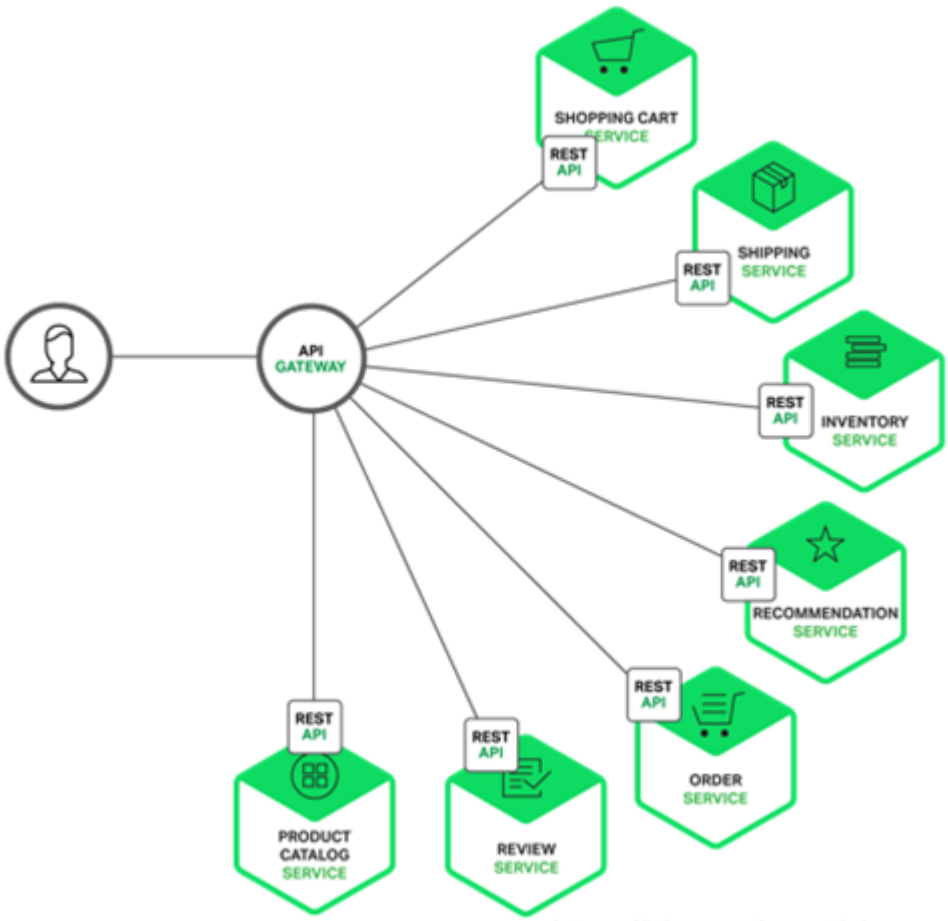


RPC（Remote Procedure Call Protocol）——远程过程调用协议，它是一种通过网络从远程计算机程序上请求服务，而不需要了解底层网络技术的协议。RPC协议假定某些传输协议的存在，如TCP或UDP，为通信程序之间携带信息数据。在OSI网络通信模型中，RPC跨越了传输层和应用层。RPC使得开发包括网络分布式多程序在内的应用程序更加容易。

2.9、微服务架构

微服务架构风格是一种将单个应用程序作为一套小型服务开发的方法，每种应用程序都在自己的进程中运行，并与轻量级机制（通常是HTTP资源API）进行通信。得益于以 Docker 为代表的容器化技术的成熟以及 DevOps 文化的的兴起，服务化的思想进一步演化，演变成我们今天所熟知的微服务。





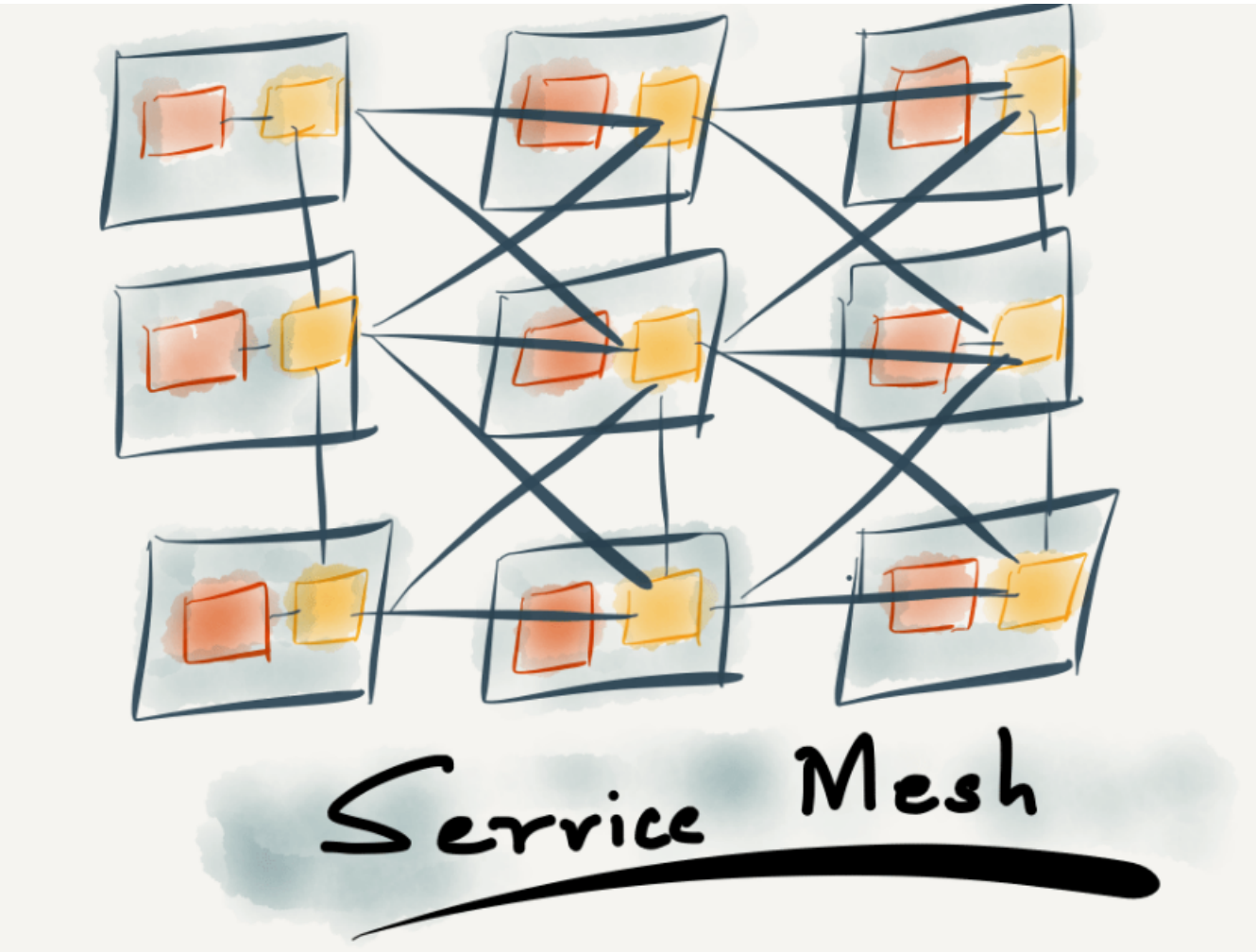
2.10 服务网格化

Service Mesh：用于处理服务间通信的基础设施层，用于在云原生应用复杂的服务拓扑中实现可靠的请求传递。在实践中，Service Mesh通常是一组与应用一起部署，但对应用透明的轻量级网络代理。

服务网格是一个基础设施层，功能在于处理服务间通信，职责是负责实现请求的可靠传递。在实践中，服务网格通常实现为轻量级网络代理，通常与应用程序部署在一起，但是对应用程序透明。

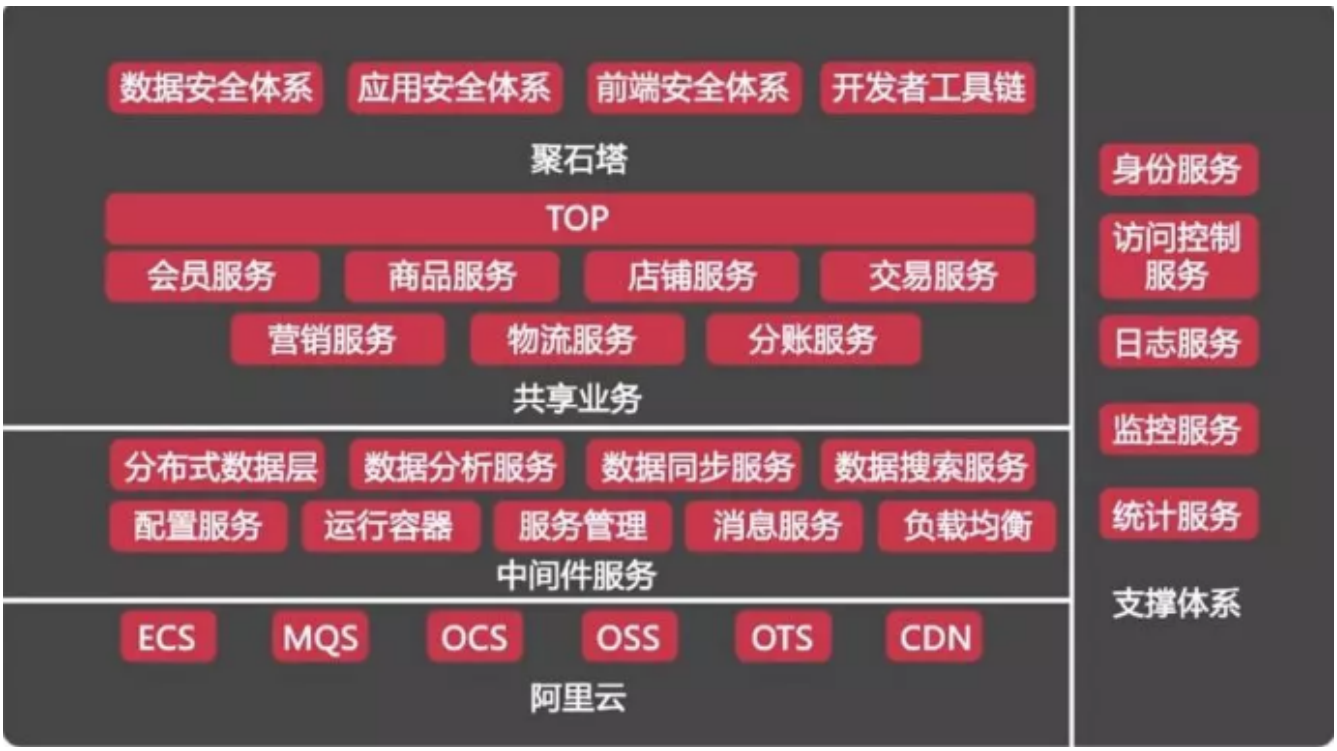
服务网格称之为下一代微服务架构标准，比如市面上流行的三套微服务架构解决方案（Apache Dubbo Zookeeper、Spring Cloud Netflix、Spring Cloud Alibaba）不难看出都是在使用各种组件组合成为我们期望的微服务架构的样子。这就直接导致了咱们目前使用的微服务架构模型都是七拼八凑的并没有一个有效的标准去界定这到底算不算微服务。

于是由 Google、IBM 和 Lyft 联合打造的 Istio 应运而生，他是 Service Mesh 的集大成者。为了实现真正的微服务架构学习 Istio 是必须的。而它又与 Kubernetes 一脉相承，因为所以 Kubernates 成为了学习Service Mesh (服务网格化)的一个基本技能

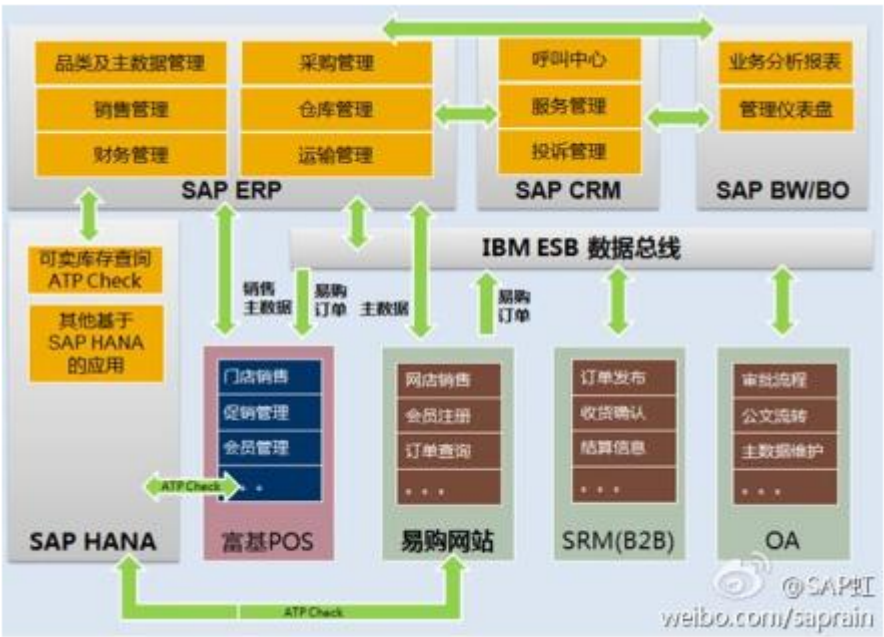


三、大型企业的架构

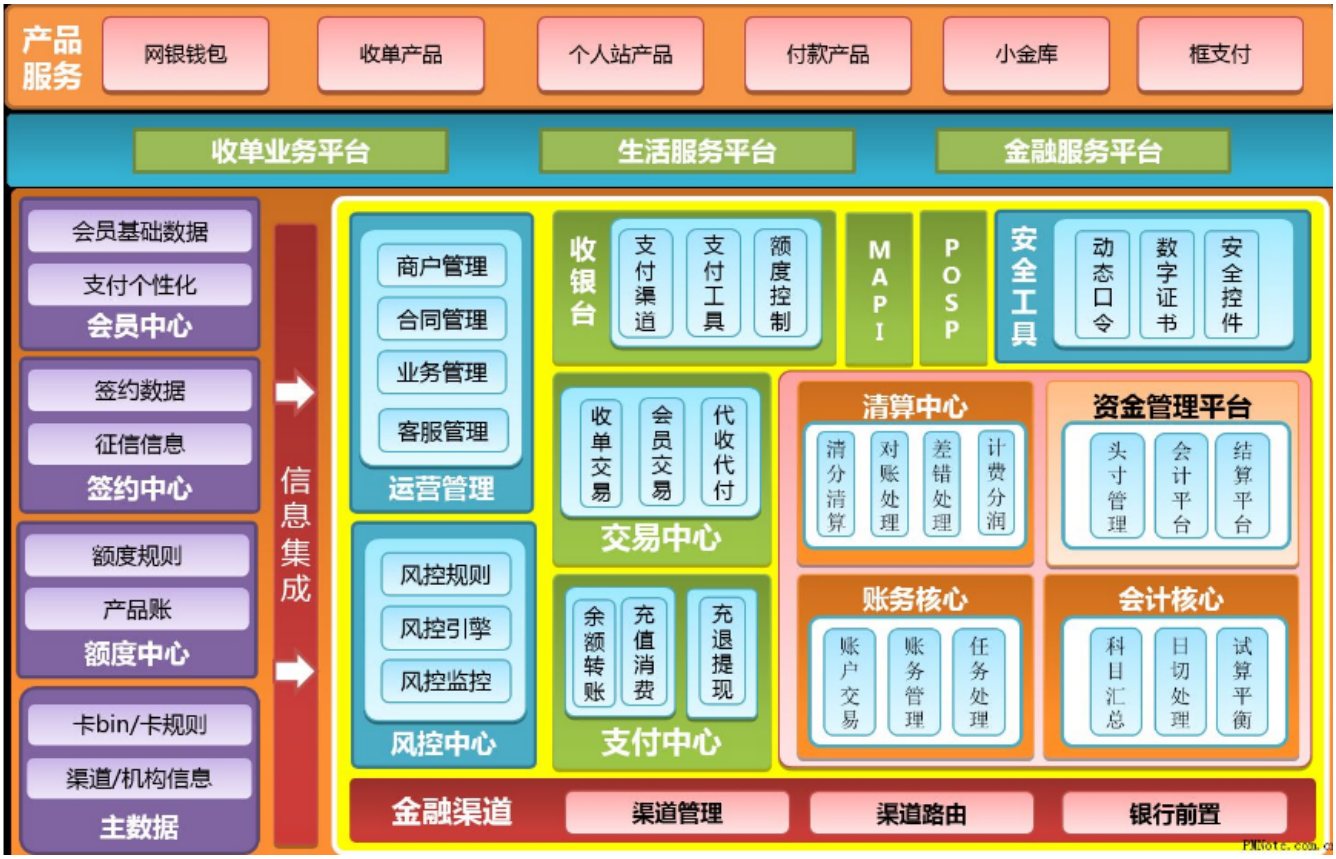
3.1 淘宝网



3.2 苏宁易购

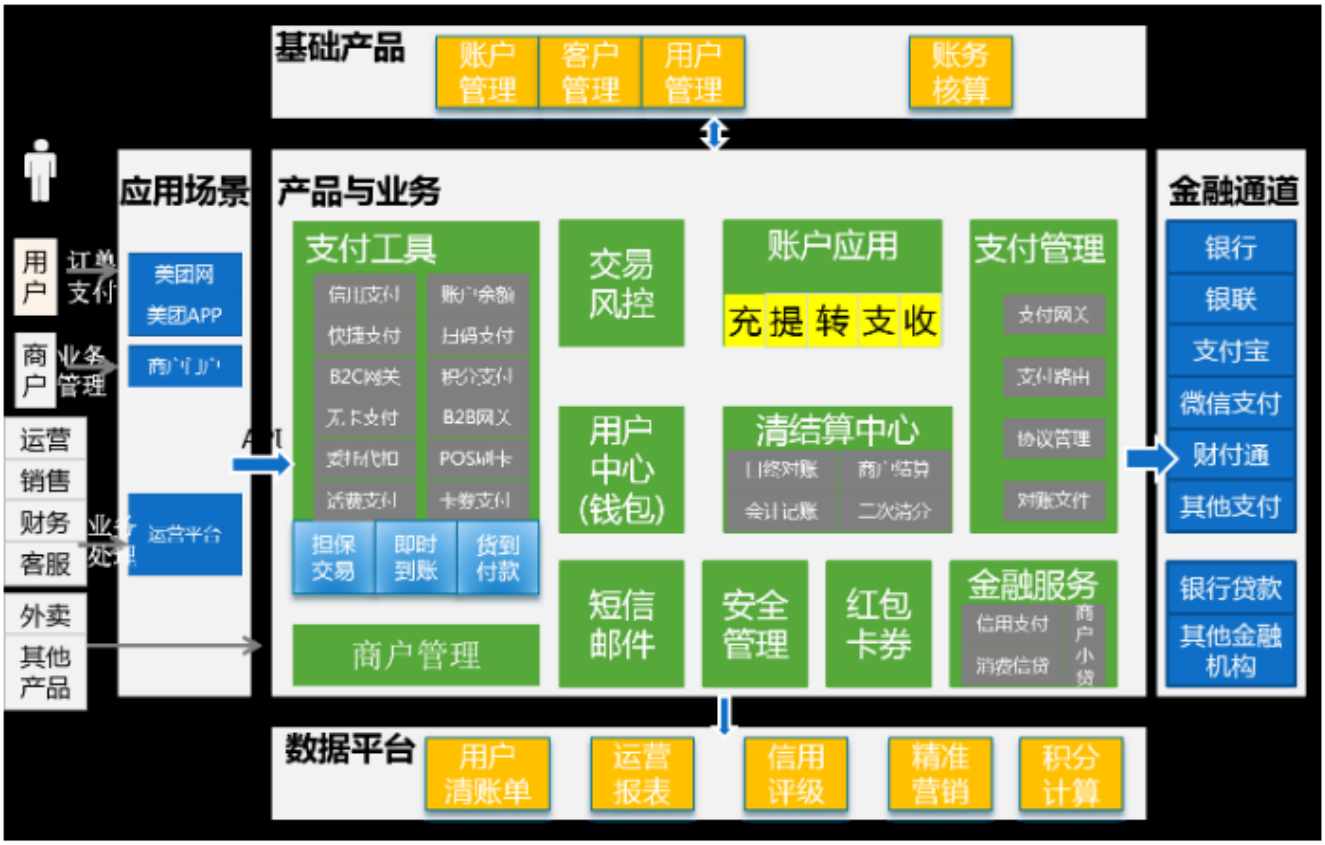


3.3 京东金融



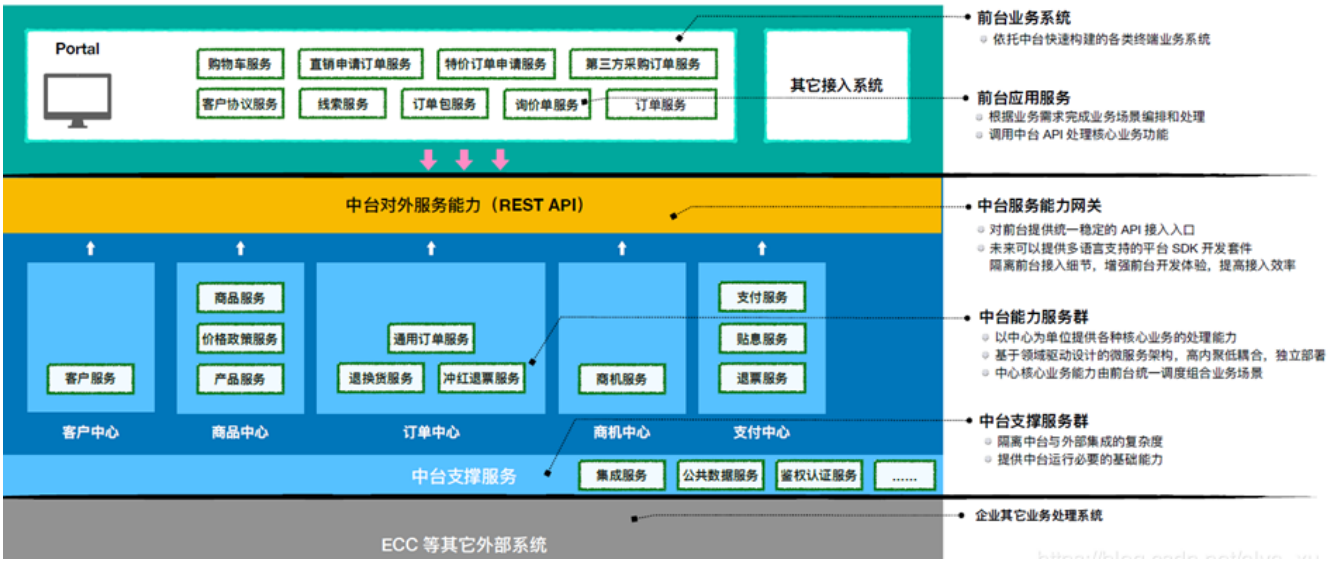
3.4 去哪儿支付平台

3.5 美团


基础产品
账户管理
客户管理
用户管理
账务核算
应用场景
产品与业务
金融通道
用户
订单支付
商户
业务管理
运营
销售
财务
客服
外卖
其他产品
运营平台
美团网
美团APP
支付工具
信用支付
账户余额
快捷支付
扫码支付
B2C网关
积分支付
无卡支付
B2B网关
受理代扣
POS刷卡
话费支付
卡券支付
担保交易
即时到账
货到付款
商户管理
交易风控
账户应用
充 提 转 支 收
支付管理
支付网关
支付路由
协议管理
对账文件
用户中心(钱包)
清结算中心
日终对账
商户结算
会计记账
二次清分
短信邮件
安全管理
红包卡券
金融服务
信用支付
消费信贷
商户小贷
银行
银联
支付宝
微信支付
财付通
其他支付
银行贷款
其他金融机构
数据平台
用户清账单
运营报表
信用评级
精准营销
积分计算

3.6 企业通用中台架构


2018云栖大会·上海峰会
THE COMPUTING CONFERENCE
阿里云
企业中台典型架构
客户
终端营销者
运营人员
后台人员
前台
营销推广
转化和交易
中台
运营支撑
后台
内部管理及后勤支持
互联网（移动）社交平台
微信
微博
云商会
粉丝论坛
体验式营销通道
自助体验
情景导购
传统营销通道
广告与媒体
电话、短信、邮件
天猫京东
自建商城
微商
B2B经销商门户
移动营销
微分销
线下网点
直营店
专卖店
在线营销
在线服务
电商订单接口
业务中台
数据中台
采购管理
生产管理
库存管理
营销管理
仓储物流
财务管理
人力资源管理
协同办公管理



https://cloud.tmall.com