

闪聚支付 第1章 讲义-项目介绍与环境搭建

1 项目介绍

1.1 项目背景

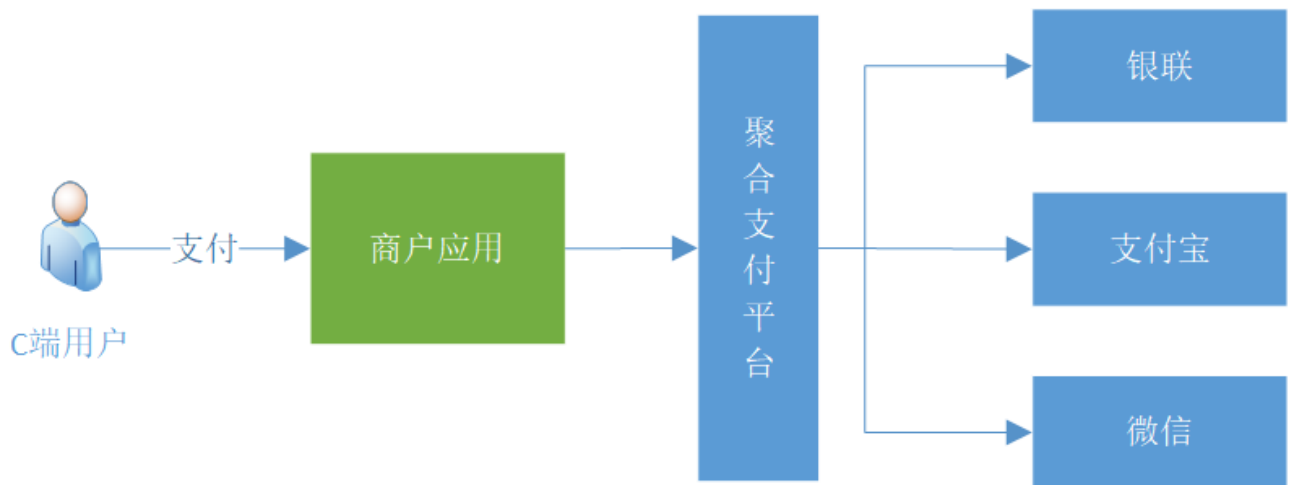
随着移动支付的盛行，商业银行、第三方支付公司、其它清算机构、消费金融公司等众多类型的机构，都在为商户提供网络（移动）支付解决方案。另一方面，用户的支付需求繁多，支付渠道已呈“碎片化”状态，并且“碎片化”程度将逐渐加深。聚合支付顾名思义就是将目前主流支付进行整合，形成第三方支付的聚合通道，也被称为“第三方支付”。

下图是聚合支付的产业结构：



聚合支付不进行资金清算，因此无需支付牌照，其只是完成支付环节的信息流转和商户运营的承载，其在集合银联、支付宝、微信等主流支付方式的基础上，帮助商户降低接入成本，提高运营效率，具有中立性、灵活性、便捷性等特点。

下图是商户通过聚合支付平台完成支付业务：



1.2 项目概述

闪聚支付是一个提供聚合支付的平台，聚合支付目前主要的做法就是线上聚合收银台(开放API)，线下C2B一码多付、线下B2C商家扫码。平台应以SaaS服务形式提供给各商户订单管理、门店管理、财务数据统计等基础服务，闪聚支付还以支付为入口，通过广告、营销、金融等服务，构建一个移动支付的全生态系统。

1.3 功能模块

平台主要包括三个模块：官网&开放平台、商户平台、运营平台，详细功能如下：



官网 & 开放平台

官网首页

产品展示

功能介绍

社区

帮助中心

产品使用快速指引

商户平台使用说明

开发者中心

开发指南

API文档

资源下载

商户平台

首页

注册

登录

基本信息

数据看板

账户管理

企业认证

企业信息

套餐购买

支付管理

创建应用

参数配置

交易总览

开启支付

门店管理

门店新增

店长分配

门店导入

员工管理

新增员工

定义角色

分配角色

优惠管理

优惠券

活动统计

运营平台

商户管理

商户检索

商户订单

商户审核

优惠管理

优惠策略

优惠券

管理员管理

新增管理员

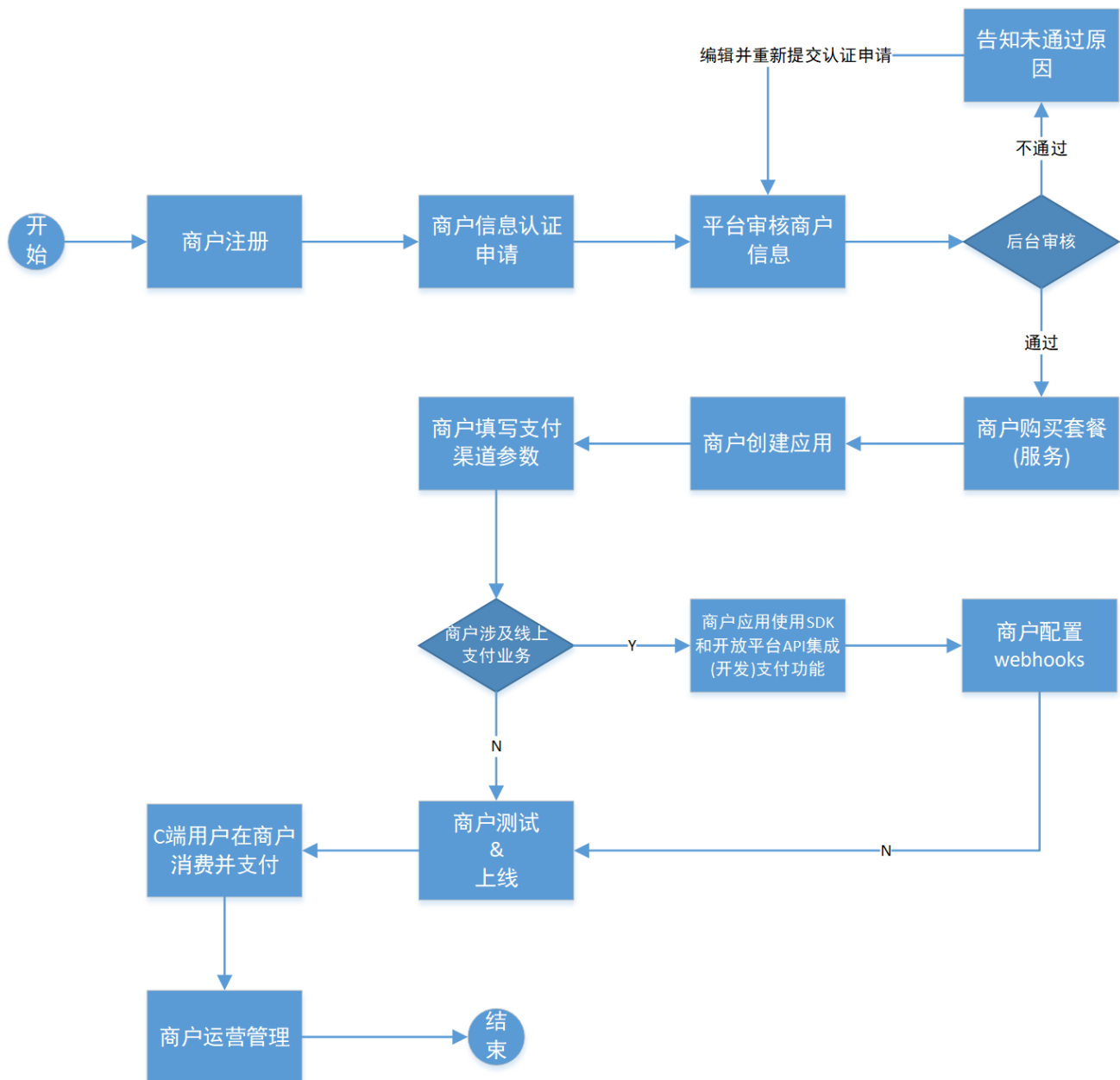
定义角色

分配角色

功能模块名称	功能说明
官网&开放平台	产品展示、功能介绍、帮助中心和开发者中心
商户平台-首页	提供商户注册、登录、基本信息查看、数据看板
商户平台-账户管理	提供商户企业资质认证、企业信息查看和套餐购买
商户平台-支付管理	为商户提供应用创建、渠道参数配置、交易总览和开启支付(生成二维码)等
商户平台-门店管理	为商户提供新增门店、店长分配、门店导入等
商户平台-员工管理	为商户提供新增员工、定义角色、分配角色等
商户平台-优惠管理	为商户提供优惠券管理、活动统计等
运营平台-商户管理	对商户进行检索、审核和订单统计
运营平台-优惠管理	为商户设置优惠策略和优惠券发送
运营平台-管理员管理	为平台提供新增管理员、定义角色和分配角色等

1.4 业务流程

平台主体业务流程如下：



1. 商户注册闪聚支付平台账号
2. 提交资质信息进行认证
3. 平台审核商户资质信息
4. 商户审核通过后，可在平台购买不同的套餐(服务)
5. 商户创建应用
6. 商户填写支付渠道参数(支付宝、微信、银联等)

如果商户要使用线上支付业务，则需要使用SDK和开放平台API集成或开发支付功能

7. 商户测试支付渠道后上线
8. C端用户在商户消费并进行二维码支付
9. 商户可通过平台进行运营管理

可以通过项目原型进一步了解项目的业务流程，包括：商户注册、C扫B、B扫C等。

参考“项目原型设计”。

2 项目的技术架构

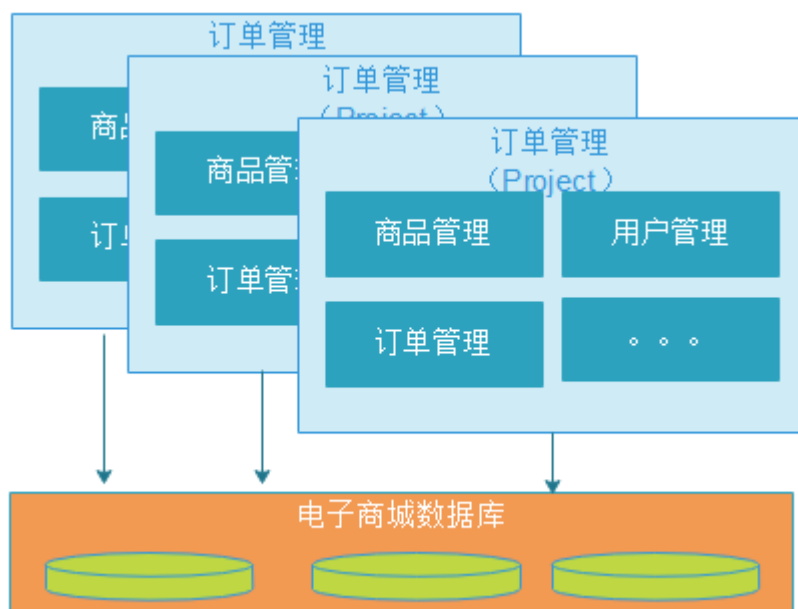
2.1 软件架构的演进

软件架构的发展经历了从单体架构、垂直架构、SOA架构到微服务架构的过程。

2.1.1 单体架构

Web应用程序发展的早期，大部分web工程师将所有的功能模块打包到一起并放在一个web容器中运行，所有功能模块使用同一个数据库。

下图是一个单体架构的电商系统：



特点：

- 1、所有的功能集成在一个项目工程中。
- 2、所有的功能打在一个war包部署到服务器。
- 3、通过部署应用集群和数据库集群来提高系统的性能。

优点：

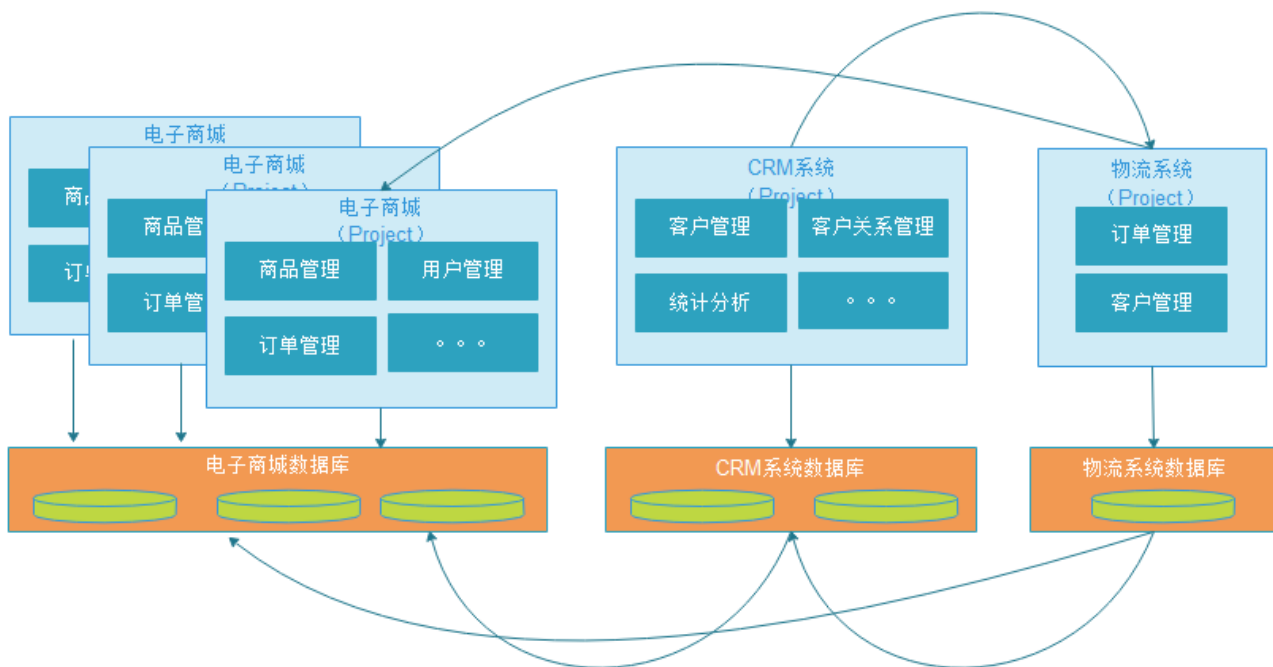
- 1、项目架构简单，前期开发成本低，周期短，小型项目的首选。
- 2、开发效率高，模块之间交互采用本地方法调用。
- 3、容易部署，运维成本小，直接打包为一个完整的包，拷贝到web容器的某个目录下即可运行。
- 4、容易测试：IDE都是为开发单个应用设计的、容易测试——在本地就可以启动完整的系统。

缺点：

- 1、全部功能集成在一个工程中，对于大型项目不易开发、扩展及维护。
- 2、版本迭代速度逐渐变慢，修改一个地方就要将整个应用全部编译、部署、启动，开发及测试周期过长。
- 3、无法按需伸缩，通过集群的方式来实现水平扩展，无法针对某业务按需伸缩。

2.1.2 分布式架构

针对单体架构的不足，为了适应大型项目的开发需求，许多公司将一个单体系统按业务垂直拆分为若干系统，系统之间通过网络交互来完成用户的业务处理，每个系统可分布式部署，这种架构称为分布式架构。



特点：

- 1、按业务垂直拆分成一个一个的单体系统，此架构也称为垂直架构。
- 2、系统与系统之间的存在数据冗余，耦合性较大，如上图中三个项目都存在客户信息。
- 3、系统之间的接口多为实现数据同步，如上图中三个项目要同步客户信息。

优点：

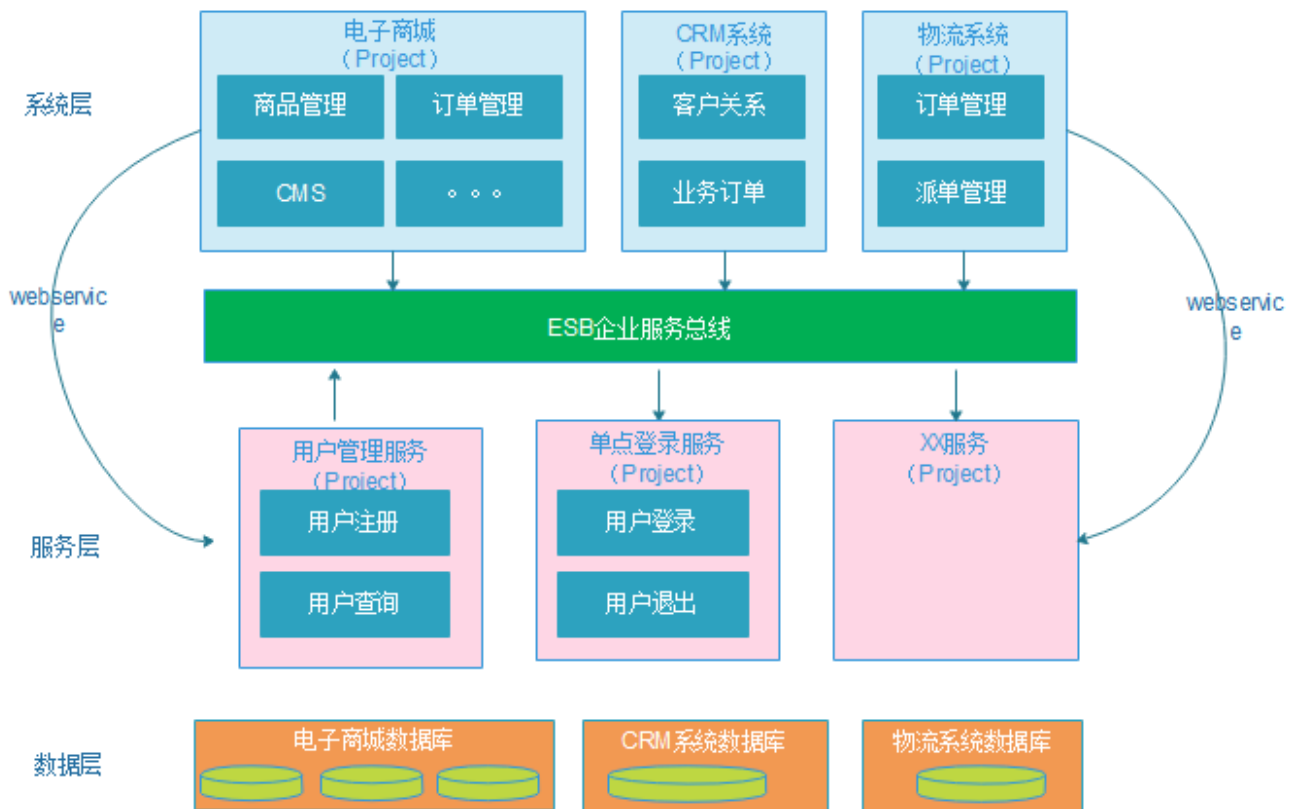
- 1、通过垂直拆分，每个子系统变成小型系统，功能简单，前期开发成本低，周期短。
- 2、每个子系统可按需伸缩。
- 3、每个子系统可采用不同的技术。

缺点：

- 1、子系统之间存在数据冗余、功能冗余，耦合性高。
- 2、按需伸缩粒度不够，对同一个子系统不同的业务无法实现，比如订单管理和用户管理。

2.1.3 SOA架构

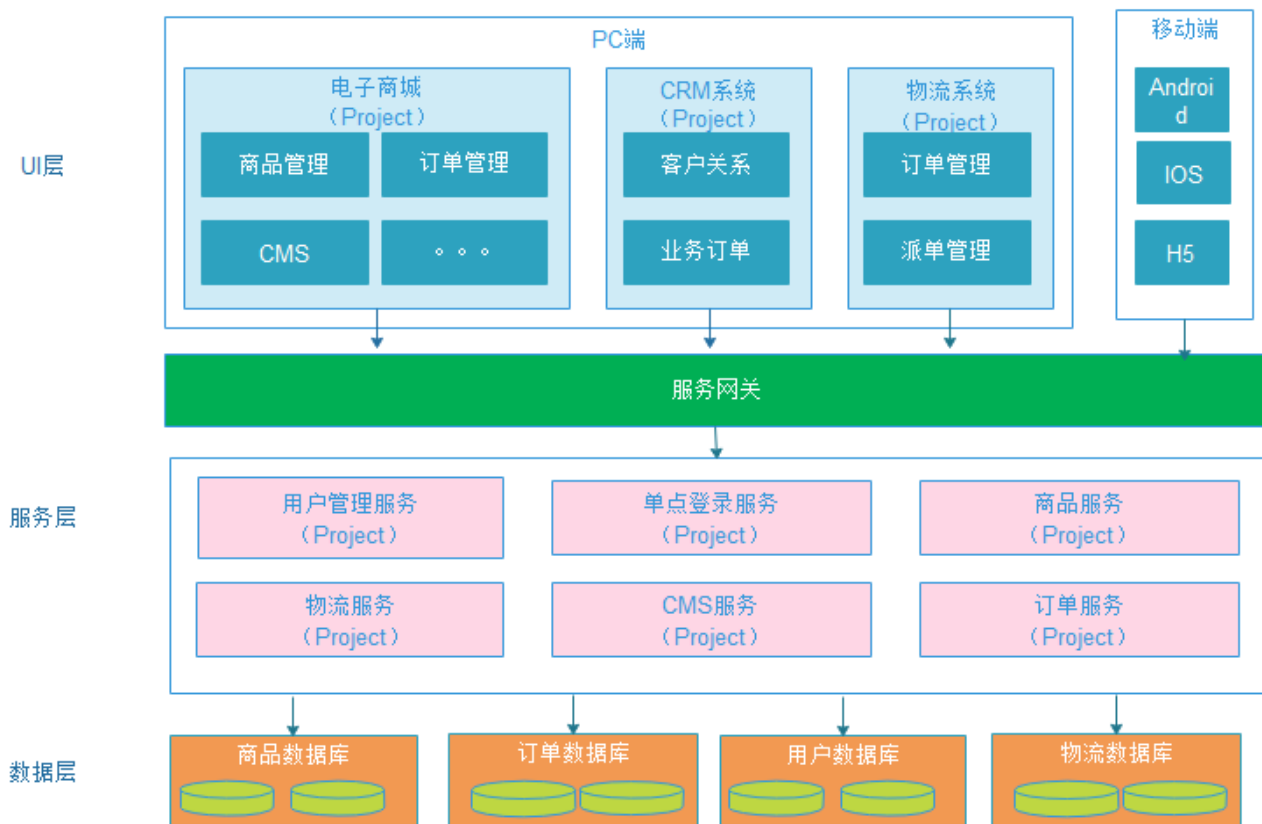
SOA是一种面向服务的架构，基于分布式架构，它将不同业务功能按服务进行拆分，并通过这些服务之间定义良好的接口和协议联系起来。



特点：1、基于SOA的架构思想，将重复公用的功能抽取为组件，以服务的方式向各各系统提供服务。2、各各系统与服务之间采用webservice、rpc等方式进行通信。3、ESB企业服务总线作为系统与服务之间通信的桥梁。优点：1、将重复的功能抽取为服务，提高开发效率，提高系统的可重用性、可维护性。2、可以针对不同服务的特点按需伸缩。3、采用ESB减少系统中的接口耦合。缺点：1、系统与服务的界限模糊，会导致抽取的服务的粒度过大，系统与服务之间耦合性高。2、虽然使用了ESB，但是服务的接口协议不固定，种类繁多，不利于系统维护。

2.1.4 微服务架构

基于SOA架构的思想，为了满足移动互联网对大型项目及多客户端的需求，对服务层进行细粒度的拆分，所拆分的每个服务只完成某个特定的业务功能，比如订单服务只实现订单相关的业务，用户服务实现用户管理相关的业务等等，服务的粒度很小，所以称为微服务架构。



特点：

- 1、服务层按业务拆分为一个一个的微服务。
- 2、微服务的职责单一。
- 3、微服务之间采用RESTful、RPC等轻量级协议传输。
- 4、有利于采用前后端分离架构。

优点：

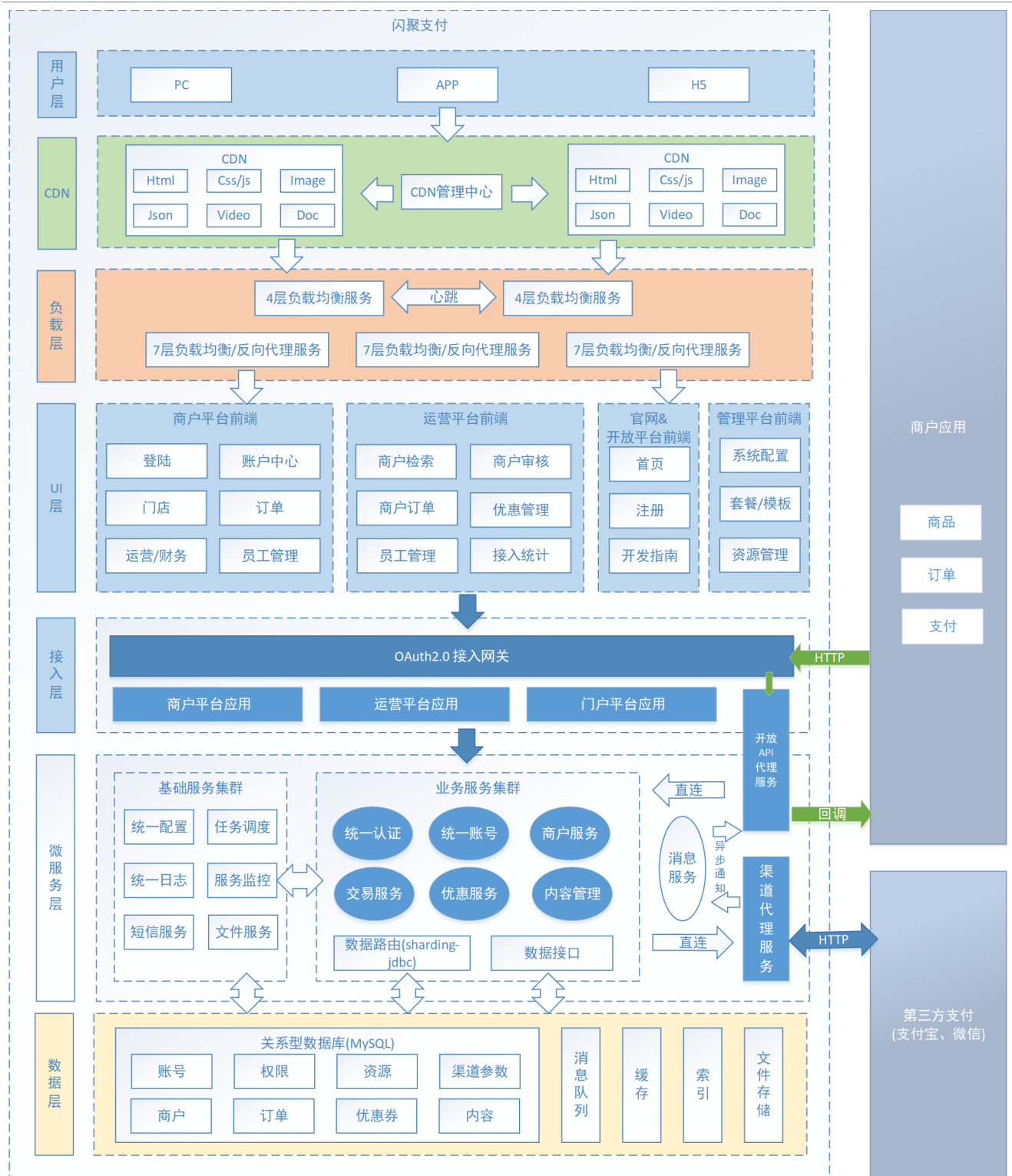
- 1、服务拆分粒度更细，有利于资源重复利用，提高开发效率。
- 2、可以更加精准的制定每个服务的优化方案，按需伸缩。
- 3、适用于互联网时代，产品迭代周期更短。

缺点：

- 1、开发的复杂性增加，因为一个业务流程需要多个微服务通过网络交互来完成。
- 2、微服务过多，服务治理成本高，不利于系统维护。

2.2 技术架构

闪聚支付采用当前流行的前后端分离架构开发，由用户层、UI层、微服务层、数据层等部分组成，为PC、H5等客户端用户提供服务。下图是系统的技术架构图：



业务流程举例：

- 1、用户可以通过PC、手机等客户端访问闪聚支付。
- 2、系统应用CDN技术，对一些图片、CSS、视频等资源从CDN调度访问。
- 3、所有的请求全部经过负载均衡器。
- 4、首先请求UI层，渲染用户界面。
- 5、商户通过平台进行注册和企业认证，UI层通过网关请求服务层，服务层完成业务处理后将数据持久化到数据层。

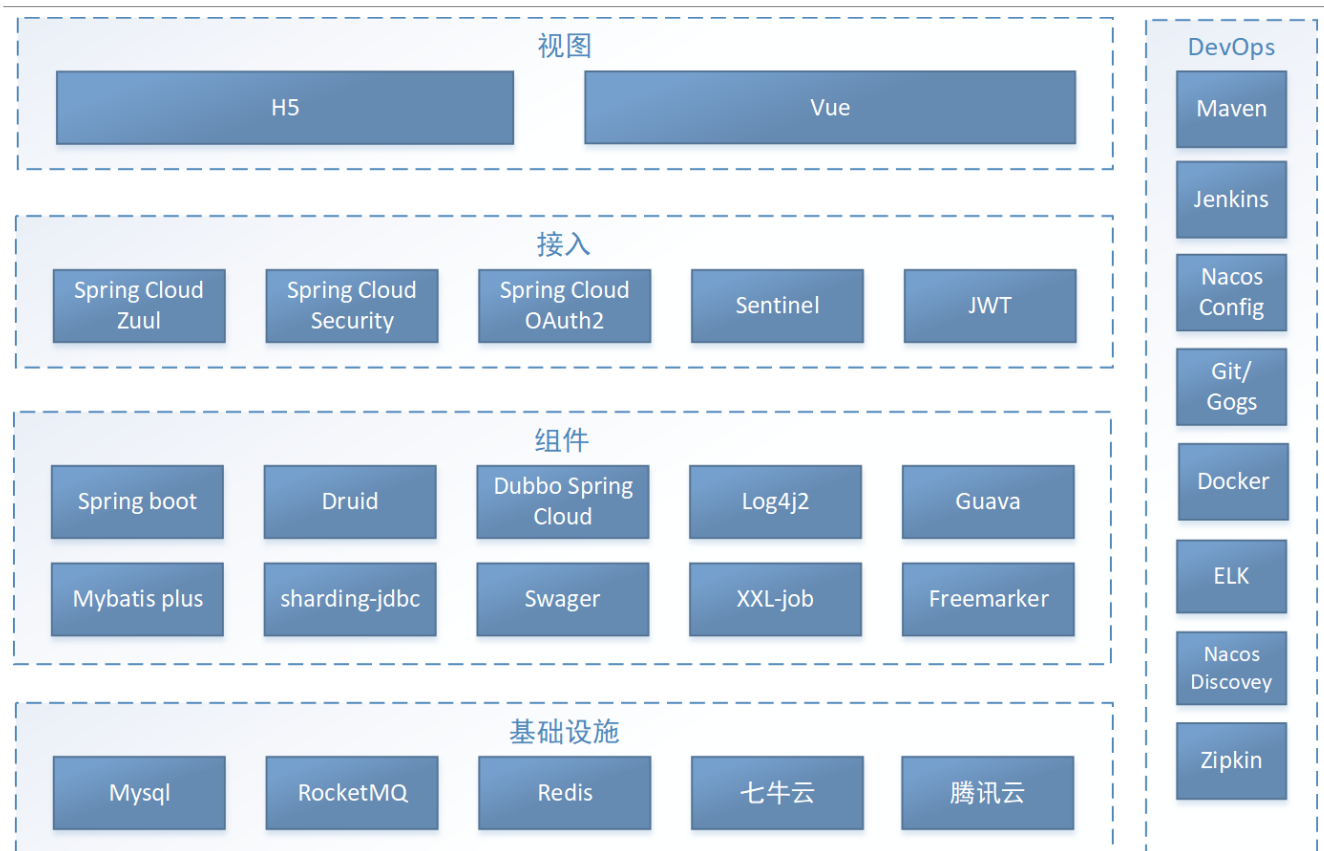
6、平台运营人员对商户信息进行审核，其系统执行流程和商户注册过程一致，UI层请求服务层业务处理，服务层查询数据层并将数据持久化到数据层。

各模块说明如下：

序号	名称	功能描述
1	用户层	用户层描述了本系统所支持的用户类型包括：pc用户、app用户、h5用户。pc用户通过浏览器访问系统、app用户通过android、ios手机访问系统，H5用户通过h5页面访问系统。
2	CDN	CDN全称Content Delivery Network，即内容分发网络，本系统所有静态资源全部通过CDN加速来提高访问速度。系统静态资源包括：html页面、js文件、css文件、image图片、pdf和ppt及doc教学文档、video视频等。
3	负载均衡	系统的CDN层、UI层、服务层及数据层均设置了负载均衡服务，上图仅在UI层前边标注了负载均衡。每一层的负载均衡会根据系统的需求来确定负载均衡器的类型，系统支持4层负载均衡+7层负载均衡结合的方式，4层负载均衡是指在网络传输层进行流程转发，根据IP和端口进行转发，7层负载均衡完成HTTP协议负载均衡及反向代理的功能，根据url进行请求转发。
4	UI层	UI层描述了系统向pc用户、app用户、h5用户提供的产品界面。根据系统功能模块特点确定了UI层包括如下产品界面类型：1) 面向商户的闪聚支付商户平台。2) 面向平台运营人员的闪聚支付运营平台。
5	微服务层	微服务层将系统服务分类两类：业务微服务、基础微服务。业务微服务：主要为商户和运营人员提供业务服务，包括统一认证、商户服务、交易服务等。基础微服务：为系统级的公共服务，不涉及具体的业务，包括文件服务、配置服务、验证码服务、调度服务等。
6	数据层	数据层描述了系统的数据存储的内容类型，持久化的业务数据使用MySQL。消息队列：存储系统服务间通信的消息，本身提供消息存取服务，与微服务层的系统服务连接。缓存：作为系统的缓存服务，存储商户信息、验证码信息、用户信息等，与微服务层的所有服务连接。文件存储：提供系统静态资源文件的分布式存储服务，文件存储服务器作为CDN服务器的数据来源，CDN上的静态资源将最终在文件存储服务器上保存多份。
7	外部系统接口	1) 支付宝、微信支付接口，本系统提供支付宝、微信两种支付接口。2) 短信接口，本系统与第三方平台对接短信发送接口。3) 文件存储，静态资源文件的存储采用第三方文件服务方式，本系统采用七牛云文件存储。4) CDN，本系统与第三方CDN服务对接，使用CDN加速服务来提高本系统的访问速度。

2.3 技术栈

下图是项目技术架构的简图，通过简图了解项目所使用的技术栈。



重点了解微服务技术栈：

闪聚支付服务端基于Spring Boot构建，采用Spring Cloud Alibaba微服务架构。

1) 基础设施

业务数据持久化采用MySQL，数据缓存采用Redis，采用RocketMQ的事务消息机制完成部分场景下的分布式事务控制，采用第三方云平台完成文件上传与分布式存储。

2) 组件

系统微服务基于SpringBoot开发，数据库连接池采用Druid，POJO构建采用Lombok，日志系统采用Log4j2，Guava工具类库，Mybatis Plus持久层接口实现，Sharding-jdbc分库分表组件，Swagger接口规范组件，XXL-job分布式任务调度组件，Sentinel限流组件等。

3) 接入

Zuul网关完成客户端认证、路由转发等功能，Dubbo RPC完成微服务远程调用，JWT提供前后端令牌管理方案。

4) 视图

平台支持H5、PC等各种前端。