# 项目简介

**【郑重声明】**

按照中国国家相关法律政策规定，不得向中国大陆境内公民提供数字资产交易服务，不得提供人民币对数字货币的兑换及支付服务。如果不予配合履行以上职责，造成的一切后果与本人无关。

# 项目简介

开源数字货币交易所，基于Java开发的货币交易所 | BTC交易所 | ETH交易所 | 数字货币交易所 | 交易平台 | 撮合交易引擎。本项目基于SpringCloudAlibaba微服务开发，可用来搭建和二次开发数字货币交易所。



**项目特色：**

1. 基于内存撮合引擎，比传统基于数据库撮合引擎更快 ；
2. 前后端分离，基于OAuth2.0 + JWT的API授权机制 ；
3. 基于SpringCloud微服务架构，扩展更容易 ;
4. 集成阿里最新的研究成功SpringCloudAlibaba ；
5. MySQL、MongoDB、Redis多种数据存储方式，只为更快 ；
6. Kafka发布订阅消息队列，让订单更快流转 ；
7. Netty秒级实时K-Line推送 ；
8. uni-app“七端”共享 ；
9. 主流币种对接区块链接口齐全，开箱即用 ；
10. 冷热钱包分离，两种提现方式，保证安全；



# 技术选型

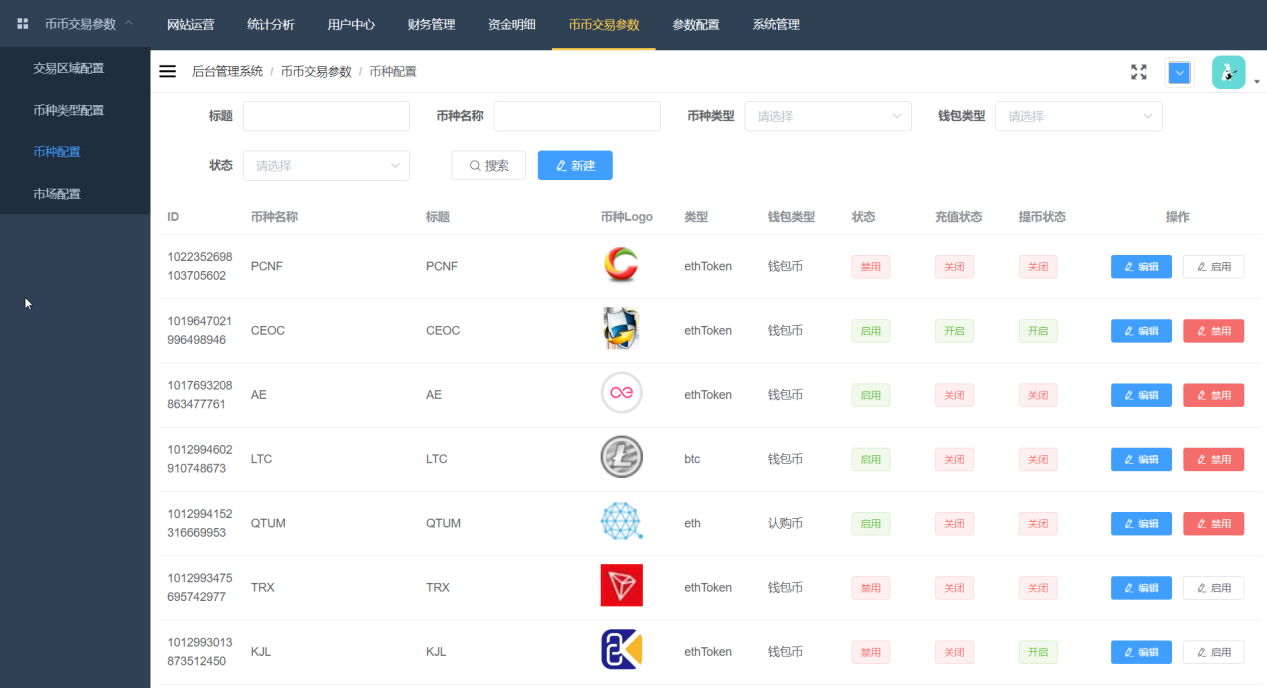
## 2.1 后台技术

|  |  |  |
| --- | --- | --- |
| 组件 | 作用 | 参考网站 |
| Spring Framework | 容器 | <http://projects.spring.io/spring-framework/> |
| Spring Boot | 开发脚手架框架 | <https://spring.io/projects/spring-cloud/> |
| Spring Cloud | 微服务框架 | <https://spring.io/projects/spring-boot/> |
| Spring Security | 安全框架 | <https://spring.io/projects/spring-security> |
| MyBatis-Plus | ORM框架 | <https://mp.baomidou.com/> |
| Nacos | 服务治理 | https://nacos.io/zh-cn/ |
| Sentinel | 服务保护 | https://sentinelguard.io/zh-cn/ |
| Seata | 分布式事务 | http://seata.io/zh-cn/ |
| MyBatis | ORM框架 | <http://www.mybatis.org/mybatis-3/zh/index.html> |
| MyBatis Generator | 代码生成 | <http://www.mybatis.org/generator/index.html> |
| PageHelper | MyBatis物理分页插件 | <http://git.oschina.net/free/Mybatis_PageHelper> |
| Druid | 数据库连接池 | <https://github.com/alibaba/druid> |
| Mongodb | 分布式文件存储数据库 | <https://www.mongodb.com/> |
| ZooKeeper | 分布式协调服务 | <http://zookeeper.apache.org/> |
| Redis | 分布式缓存数据库 | <https://redis.io/> |
| Redisson | Redis客户端 | <https://redisson.org/> |
| Jetcache | 缓存框架 | [https://github.com/alibaba/jetcache](https://github.com/alibaba/jetcache/) |
| RabbitMQ | 消息队列 | <https://www.rabbitmq.com/> |
| Kafka | 消息队列 | <http://kafka.apache.org/> |
| Disruptor | 并发框架 | <https://lmax-exchange.github.io/disruptor/> |
| FastDFS | 自建分布式文件系统 | <https://github.com/happyfish100/fastdfs> |
| Log4J | 日志组件 | <http://logging.apache.org/log4j/1.2/> |
| Swagger2 | 接口测试框架 | <http://swagger.io/> |
| Lombok | 简化编码插件 | <https://projectlombok.org/> |
| Cloud Alibaba OSS  Qiniu | 云存储 | <https://www.aliyun.com/product/oss/> <http://www.qiniu.com/> |
| FastJson & Gson | 数据序列化 | <https://github.com/alibaba/fastjson> |
| Jenkins | 持续集成工具 | <https://jenkins.io/index.html> |
| Maven | 项目构建管理 | <http://maven.apache.org/> |
| Tio | 实时推送 | <https://gitee.com/tywo45/t-io> |
| Netty | 实时推送 | https://netty.io/ |
| Akka | 异步并发计算 | https://akka.io/ |

## 2.2 前端技术

### 2.2.1 后台管理系统

Vue + ElementUI + Axios + Xlsx + Showdown + Screenfull



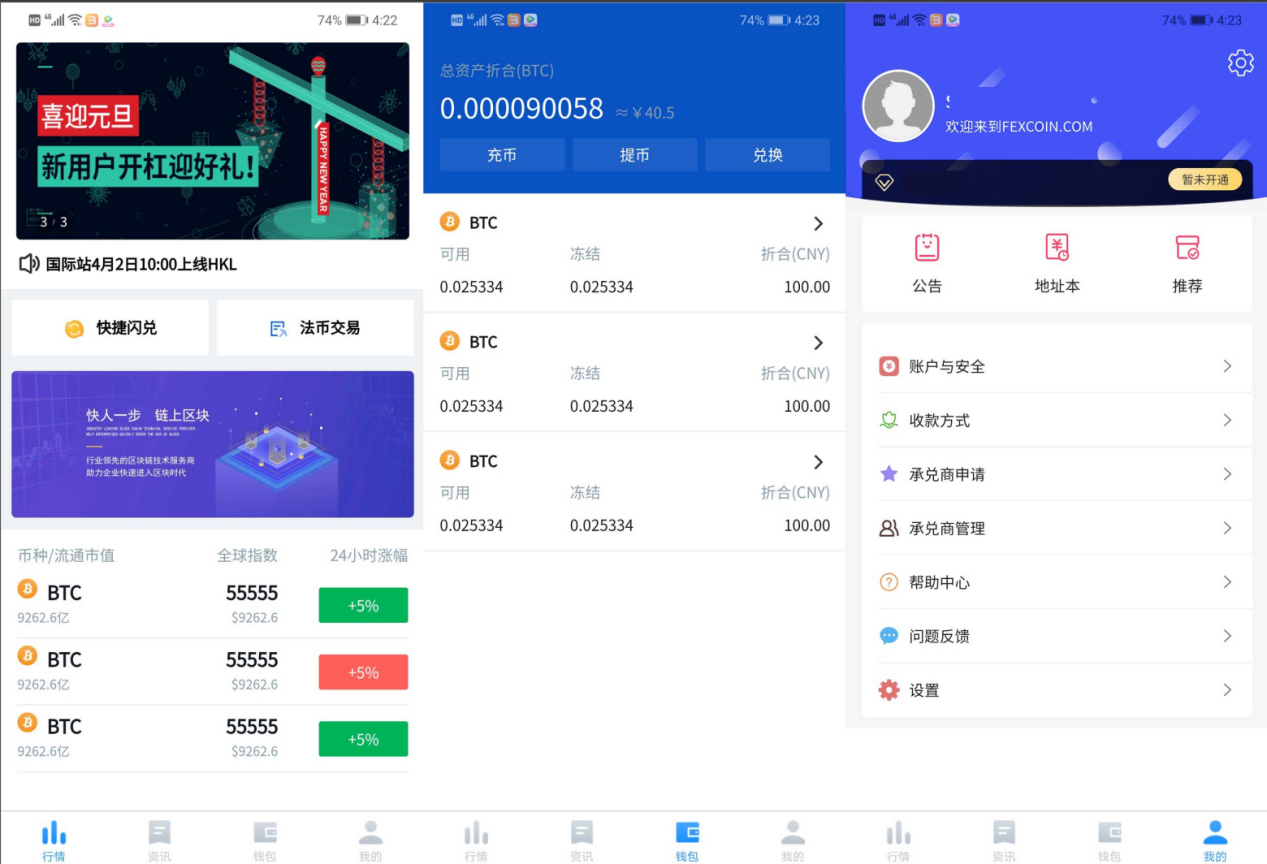
### 2.2.2 前台系统

Vue + ElementUI + Axios + Stompjs + Zip + vue-i18n



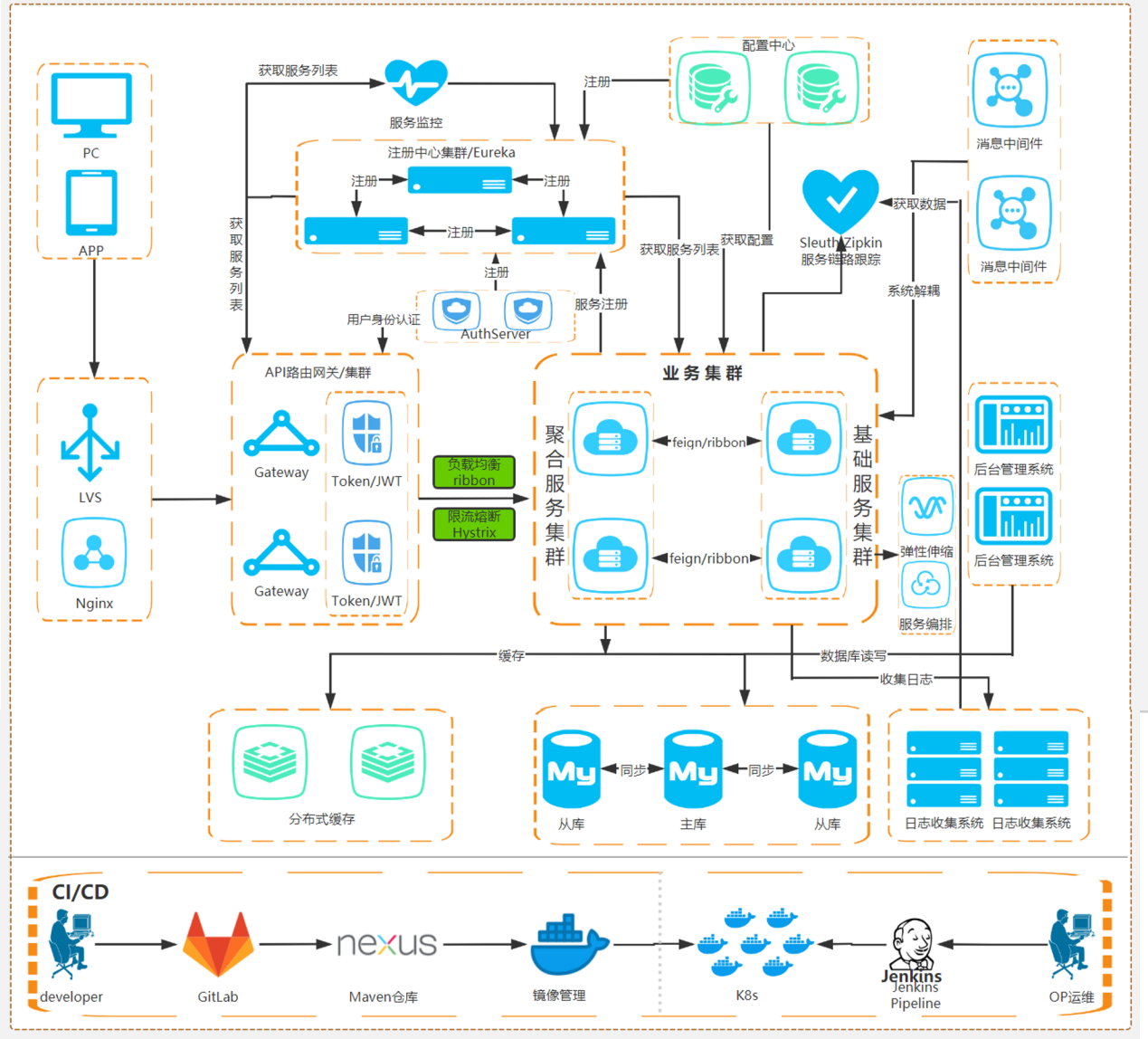
### 2.2.3 移动全栈

**uni-app**



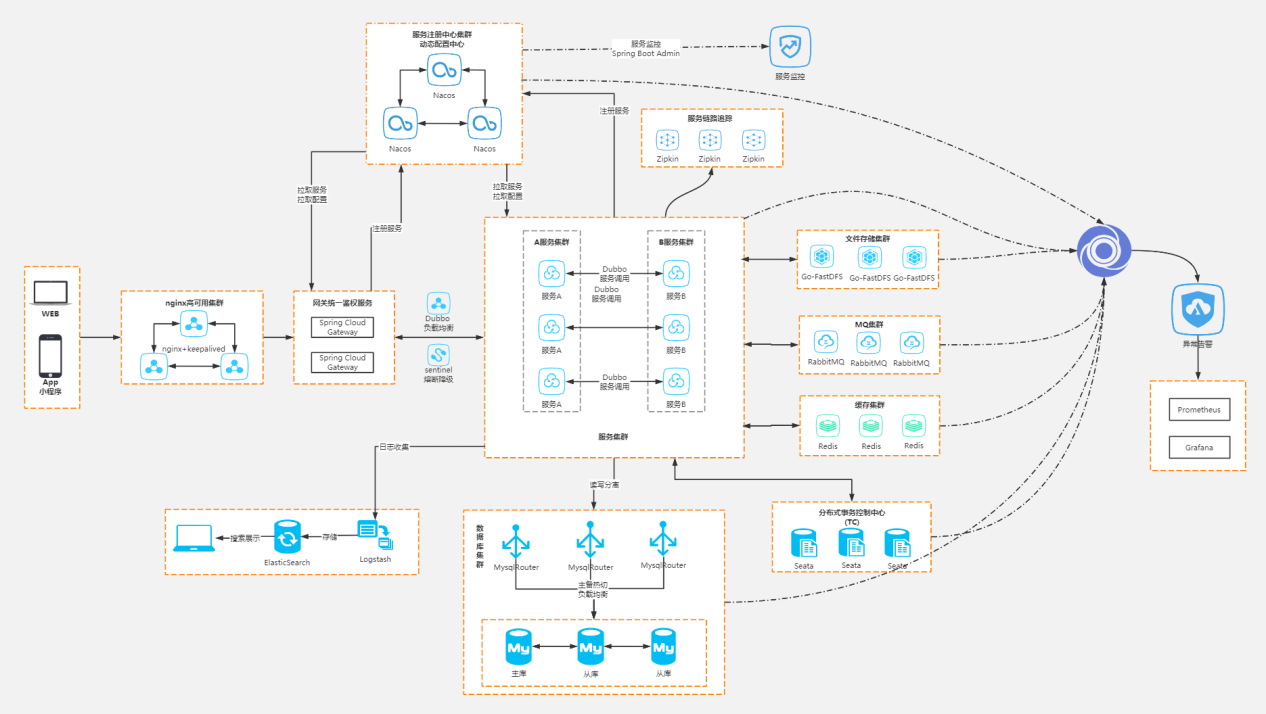
# 关于Spring Cloud

Spring Cloud是一系列框架的有序集合。它利用Spring Boot的开发便利性巧妙地简化了分布式系统基础设施的开发，如服务发现注册、配置中心、消息总线、负载均衡、断路器、数据监控等，都可以用Spring Boot的开发风格做到一键启动和部署。Spring Cloud并没有重复制造轮子，它只是将各家公司开发的比较成熟、经得起实际考验的服务框架组合起来，通过Spring Boot风格进行再封装屏蔽掉了复杂的配置和实现原理，最终给开发者留出了一套简单易懂、易部署和易维护的分布式系统开发工具包。 一般而言，一个完整的SpringCloud框架应该如下图所示：



# 关于Spring Cloud Alibaba

Spring Cloud Alibaba 致力于提供微服务开发的一站式解决方案。此项目包含开发分布式应用微服务的必需组件，方便开发者通过 Spring Cloud 编程模型轻松使用这些组件来开发分布式应用服务。

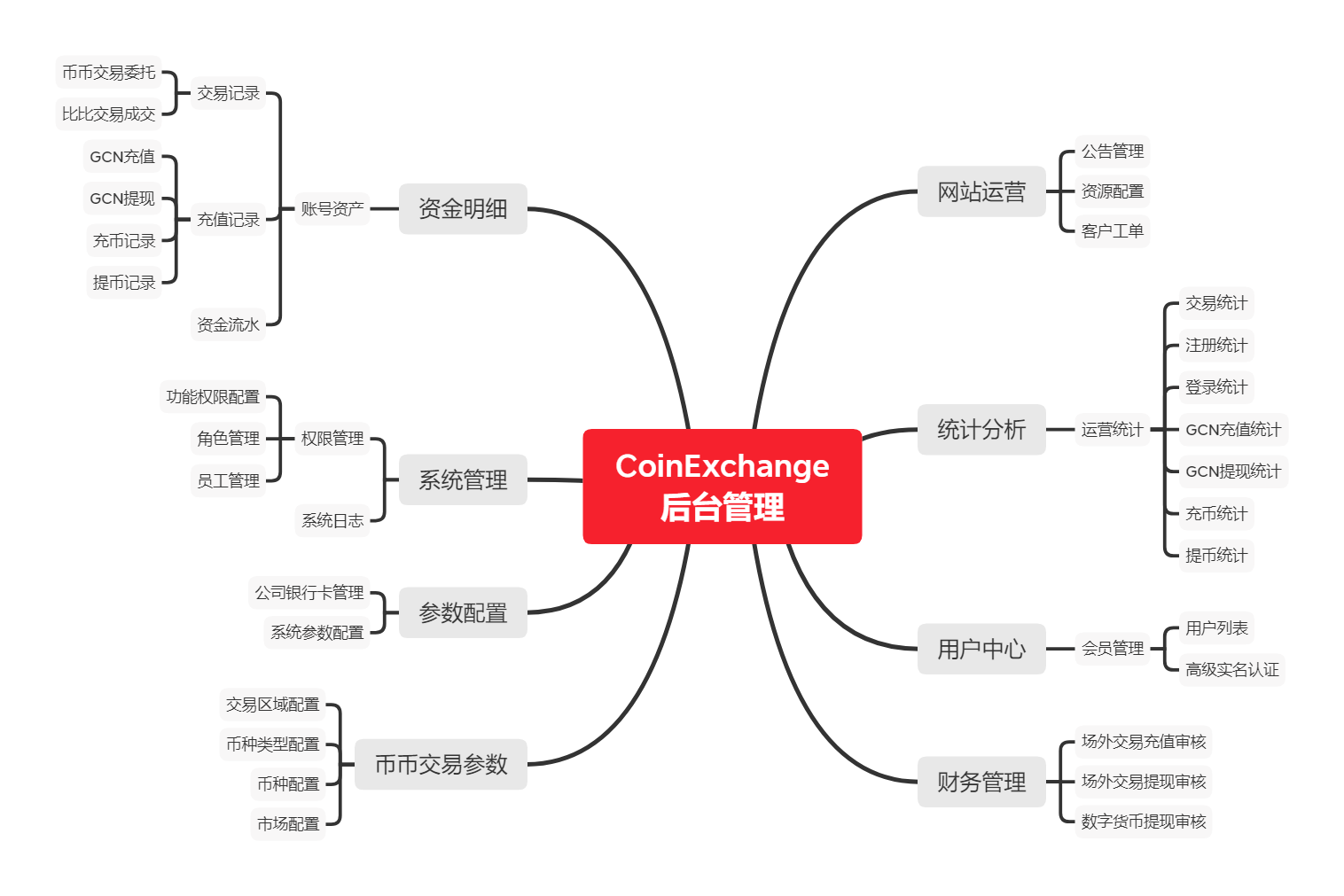


**组件：**

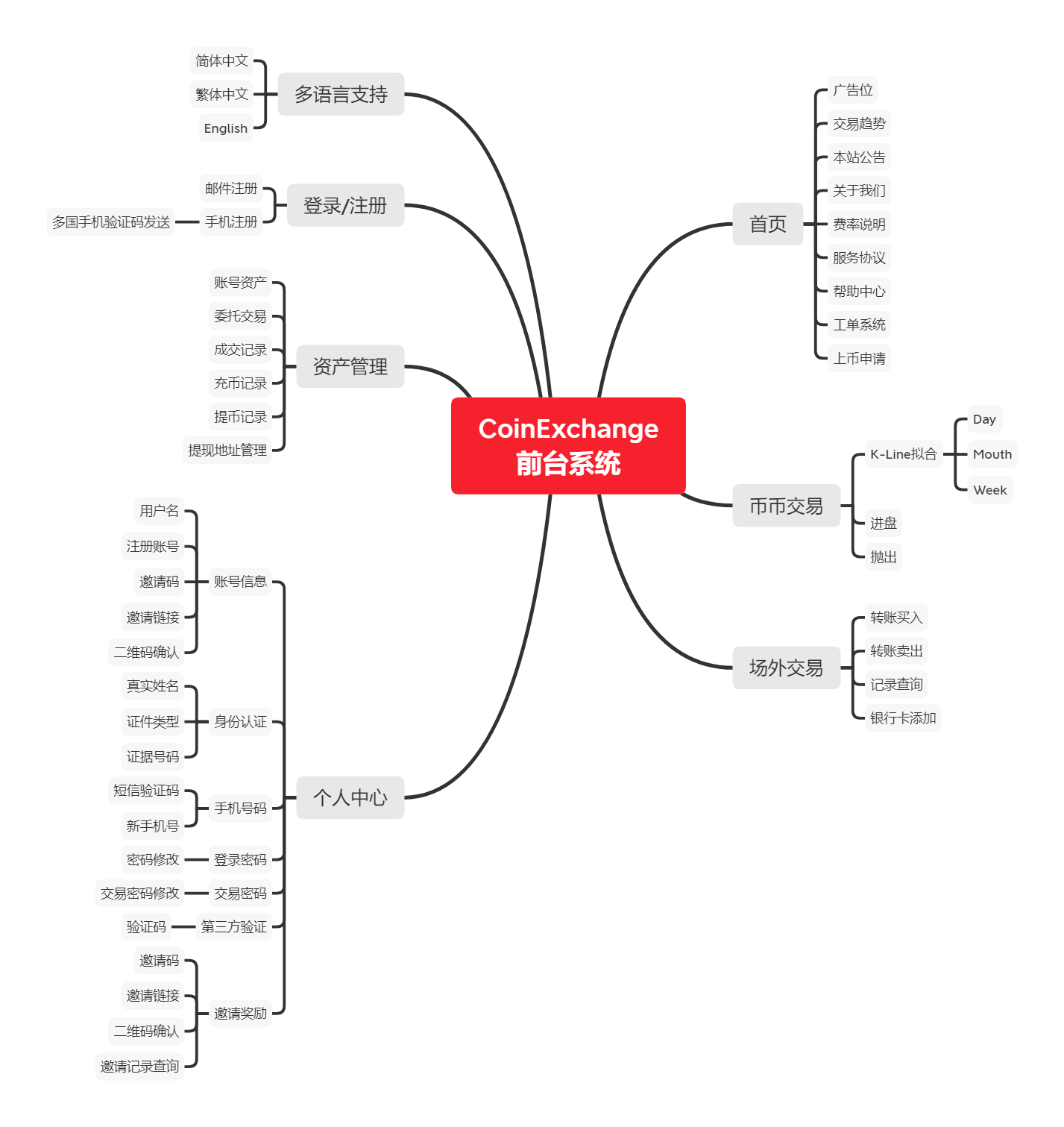
* Sentinel：把流量作为切入点，从流量控制、熔断降级、系统负载保护等多个维度保护服务的稳定性。
* Nacos：一个更易于构建云原生应用的动态服务发现、配置管理和服务管理平台。
* RocketMQ：一款开源的分布式消息系统，基于高可用分布式集群技术，提供低延时的、高可靠的消息发布与订阅服务。
* Seata：阿里巴巴开源产品，一个易于使用的高性能微服务分布式事务解决方案。
* Alibaba Cloud OSS: 阿里云对象存储服务（Object Storage Service，简称 OSS），是阿里云提供的海量、安全、低成本、高可靠的云存储服务。您可以在任何应用、任何时间、任何地点存储和访问任意类型的数据。
* Alibaba Cloud SMS: 覆盖全球的短信服务，友好、高效、智能的互联化通讯能力，帮助企业迅速搭建客户触达通道。

# 功能介绍和演示

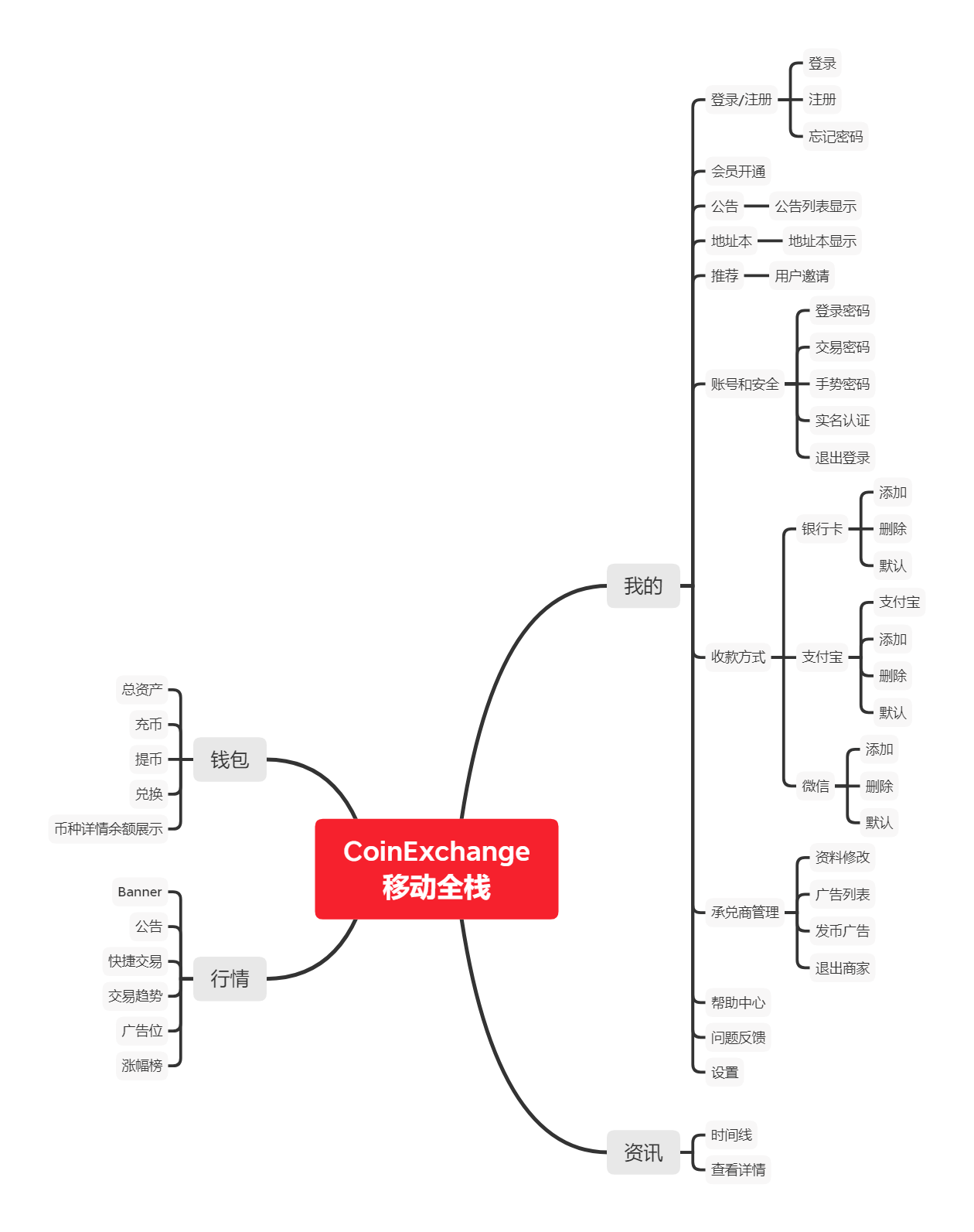
## 5.1 后台管理系统



## 5.2 前台系统



## 5.3 移动全栈



# 架构设计

## 6.1 模块划分

### 6.1.1 撮合引擎服务

分布式内存撮合系统，使用Raft协议自建集群，实现数据复制、集群选举等，并自研分布式币种路由策略，解决多币之间数据隔离，线程竞争问题；当主机挂掉时，备机能快速感知并替换原主机状态继续提供服务，当集群扩容时，也实现数据分片的迁移和再分配。在部署时，我们可以采用单一分组，也可以采用 MULTI-RAFT-GROUP。我们还使用Nosql数据库mongo作为recovery的兜底方案，当撮合引擎所有节点都进行重新实例化时，那么就会走该兜底方案。

### 6.1.2 行情服务

* OpenAPI

对外提供OpenAPI(开放平台)，我们所有API的设计遵循REST 设计标准，使用改造完后的swagger生成更符合国人习惯的测试平台；

* K-Line

K线为专业K线，K线可自定义日K、1分、3分、5分、15 分、30分、10小时、2小时、4小时、6小时、12小时、1周等。价格平均线可以自定义：MA5、MA10、MA30及其他个性化配置。技术指标包括：MACD、 KDJ、RSI、DMI、OBV、BOLL、SAR、DMA、TRIX、BRAR、VR、EMV、WR、ROC、MTM、PSY。

### 6.1.3 推送服务

推送服务为实时推送，数据的获取来自Kafka，多线程订阅Kafka的分组，底层通过修改spring-boot-stater-websocket让其支持Netty，并测试了TIO的推送方案，当客户端压力变大时，可通过Spring Cloud Gateway 和 Nginx 实现负载均衡。经测试，客户端无卡顿或数据丢失的情况。

### 6.1.4 交易服务

币种获取、区域数据、币币交易市场行情，币币交易市场深度等各种交易数据展示，底层采用Redis做高速缓存，并使用Sentinel实现了服务的降级和流控。

### 6.1.5 用户中心服务

提供用户账号信息、身份验证、注册登陆、邀请分级奖励、第三方验证等功能；

### 6.1.6 监控服务

* Boot-Admin ：

实现了服务上下线的邮件/短信/微信通知，显示微服务 name、id 和 version，显示在线状态，Loggers 的日志级别管理，Threads 线程管理，Environment 管理等；

* sentinel-dashboard：

提供一个轻量级的控制台，它提供机器发现、单机资源实时监控、集群资源汇总，以及规则管理的功能；

* Zipkin：

提供分布式链路调用监控系统，聚合各业务系统调用延迟数据，达到链路调用监控跟踪，以及服务直接依赖关系的计算。

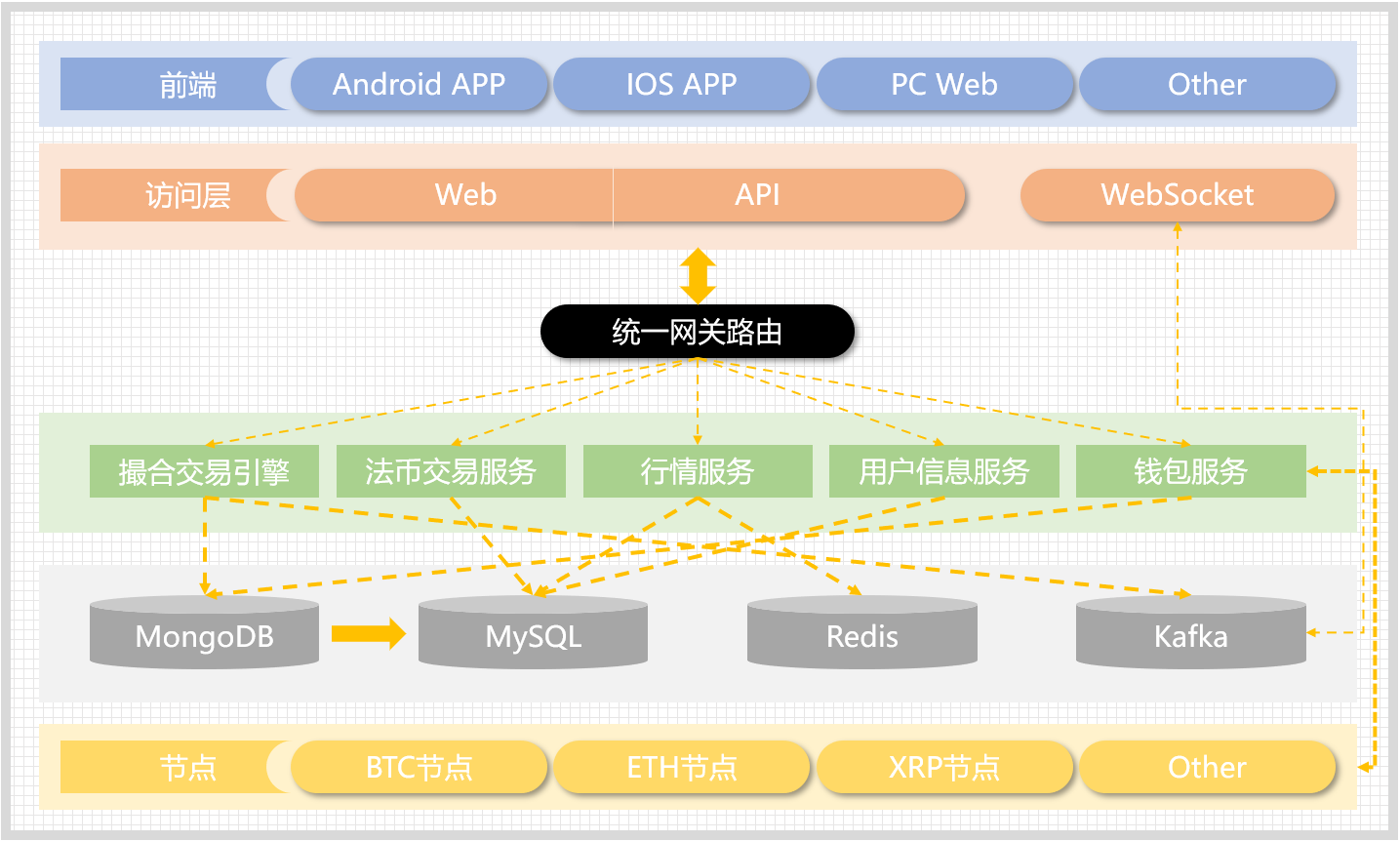
### 6.1.7 钱包服务

采用了业界领先的冷热隔离充值钱包系统、多重签名、高防护DDOS攻击系统等领先技术，分钟级日志分析监控等手段，充分保证客户资产安全。

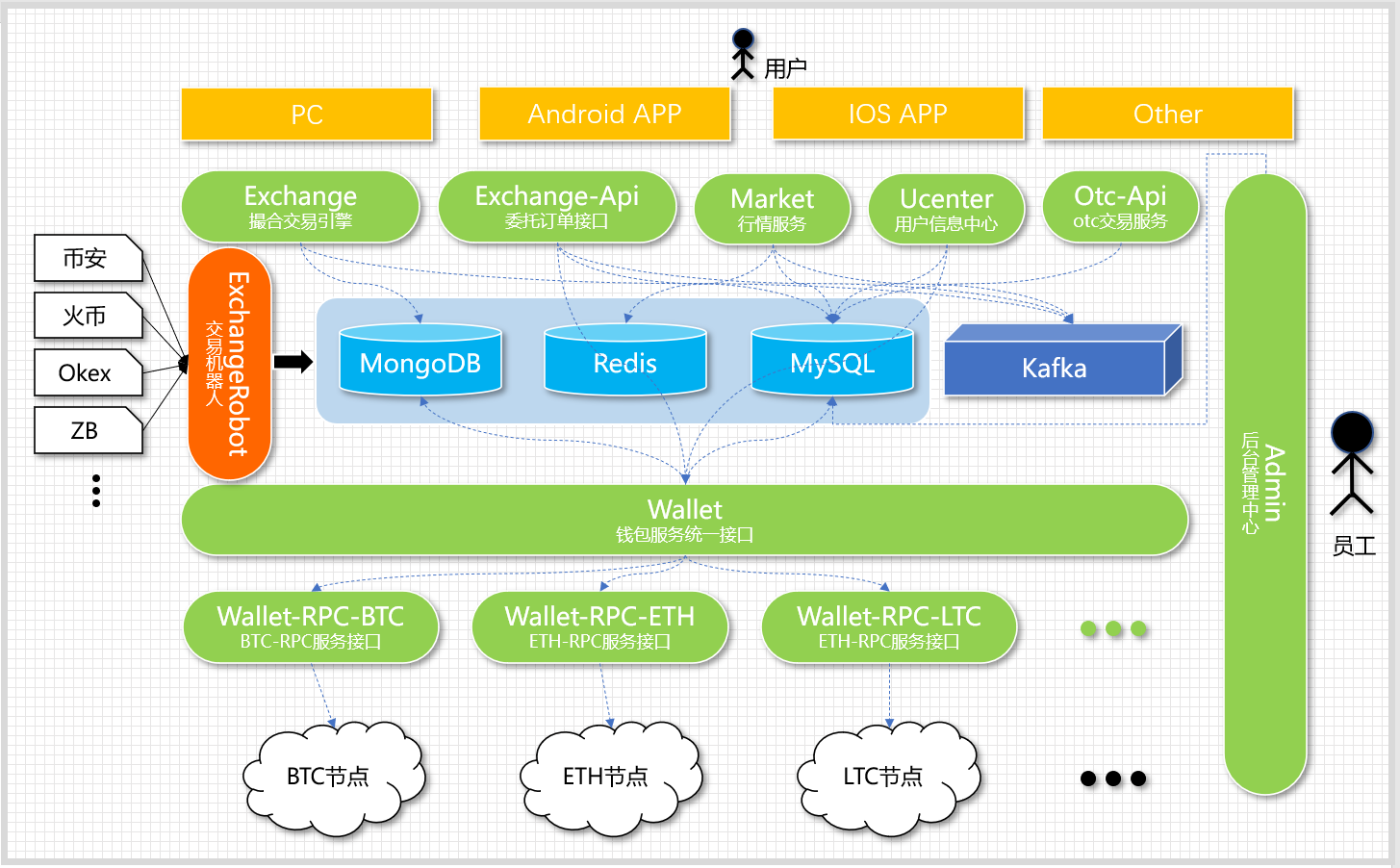
## 6.2 设计要求



## 6.3 整体架构



# 逻辑架构



# 部署架构

