## 服务通信

### 3.1 服务间调用

- 采用OpenFeign进行服务间的同步调用

- 使用RabbitMQ进行异步消息通信

- 统一的接口规范和错误码定义

### 3.2 数据一致性

- 采用Seata确保分布式事务一致性

- 关键业务采用最终一致性方案

- 使用分布式锁处理并发

## 四、安全设计

### 4.1 认证授权

- 统一认证中心

- JWT Token认证

- 基于RBAC的权限模型

- 数据权限控制

### 4.2 安全防护

- 网关统一鉴权

- 服务间调用鉴权

- 敏感数据加密

- SQL注入防护

- XSS防护

## 五、可用性设计

### 5.1 服务高可用

- 服务注册发现

- 负载均衡

- 熔断降级

- 限流保护

### 5.2 监控告警

- 服务监控

- 链路追踪

- 日志收集

- 性能监控

- 告警通知

## 六、部署架构

### 6.1 环境规划

- 开发环境

- 测试环境

- 预发环境

- 生产环境

### 6.2 容器化部署

- 基于Docker容器化部署

- 使用Docker Compose编排

- 支持K8s集群部署

- Jenkins自动化部署

## 七、扩展性设计

### 7.1 水平扩展

- 服务无状态设计

- 数据库读写分离

- 缓存集群

- 消息集群

### 7.2 垂直扩展

- 业务模块化

- 数据分库分表

- 多租户支持

- 插件化架构

## 八、开发规范

### 8.1 项目规范

- 统一的项目结构

- 统一的命名规范

- 统一的接口规范

- 统一的日志规范

### 8.2 开发流程

- 代码版本控制

- 分支管理策略

- 代码审查流程

- 持续集成部署