秒杀

概要设计

|  |  |
| --- | --- |
| 文件标识： |  |
| 应用部门： | 研究院-项目研发组 |
| 保存期限： |  |

**文档修订历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 版本 | 说明 | 作者 | 审批人 |
| 2020-11-20 | V1.01.001 | 新建 | 沈坤林 |  |
|  |  |  |  |  |

目录

[1. 引言 4](#_Toc1262)

[1.1. 编写目的 4](#_Toc28845)

[1.2. 项目背景 4](#_Toc14492)

[1.3. 定义 4](#_Toc381)

[1.4. 参考资料 4](#_Toc15933)

[2. 业务概述 5](#_Toc5512)

[2.1. 需求概述 5](#_Toc11250)

[系统功能 6](#_Toc1283)

[2.1.1. 核心业务流程 6](#_Toc4433)

[2.1.2. 功能模块划分 7](#_Toc8345)

[2.2. 设计约束 8](#_Toc9753)

[3. 系统结构 9](#_Toc19398)

[3.1. 技术架构图 9](#_Toc1960)

[3.1.1. 项目目录结构说明 11](#_Toc7928)

[3.2. 项目部署 12](#_Toc17173)

[3.2.1. 服务部署 12](#_Toc30659)

[3.2.2. 运行环境部署 14](#_Toc6424)

[3.2.3. DevOps环境部署 14](#_Toc20396)

[4. 技术体系 15](#_Toc4643)

[4.1. 技术栈 15](#_Toc13911)

[4.2. 技术亮点 17](#_Toc27740)

[4.2.1. 基础 17](#_Toc29214)

[4.2.2. 实战 17](#_Toc31320)

[4.2.3. 运维 17](#_Toc8789)

[5. 业务逻辑设计 18](#_Toc204)

[5.1. 主要业务逻辑列表 18](#_Toc22734)

[5.2. 主要业务逻辑设计 18](#_Toc21126)

[5.2.1. 用户登录（略） 18](#_Toc3159)

[5.2.2. 实时热点数据收集 19](#_Toc9580)

[5.2.3. 秒杀抢单 20](#_Toc5189)

[6. 内部接口 22](#_Toc12383)

[6.1. 接口列表 22](#_Toc7505)

[7. 数据库设计 24](#_Toc24872)

[7.1. 实体关系图(E-R) 24](#_Toc19019)

# 引言

## 编写目的

此文档是针对秒杀系统的调研、分析后的设计产出。为后续项目开发及课程研发提供总体性支撑。

本文档阅读对象包括：

* 系统设计人员
* 系统开发工程师
* 项目经理
* 课程研发人员
* 课程讲师

## 项目背景

随着互联网以及网购的普及，网购用户越来越多，而秒杀在电商项目中无论是业务设计还是技术实现都有举足轻重的地位，我们的学员如果掌握电商秒杀的技术，将更有利于学员高薪就业。。

## 定义

| 术语 | 描述 |
| --- | --- |
| 微服务架构 | 微服务架构（Microservice Architecture）是一种架构概念，旨在通过将功能分解到各个离散的服务中以实现对解决方案的解耦。你可以将其看作是在架构层次而非获取服务的类上应用很多SOLID原则。微服务架构是个很有趣的概念，它的主要作用是将功能分解到离散的各个服务当中，从而降低系统的耦合性，并提供更加灵活的服务支持。 |

## 参考资料

略

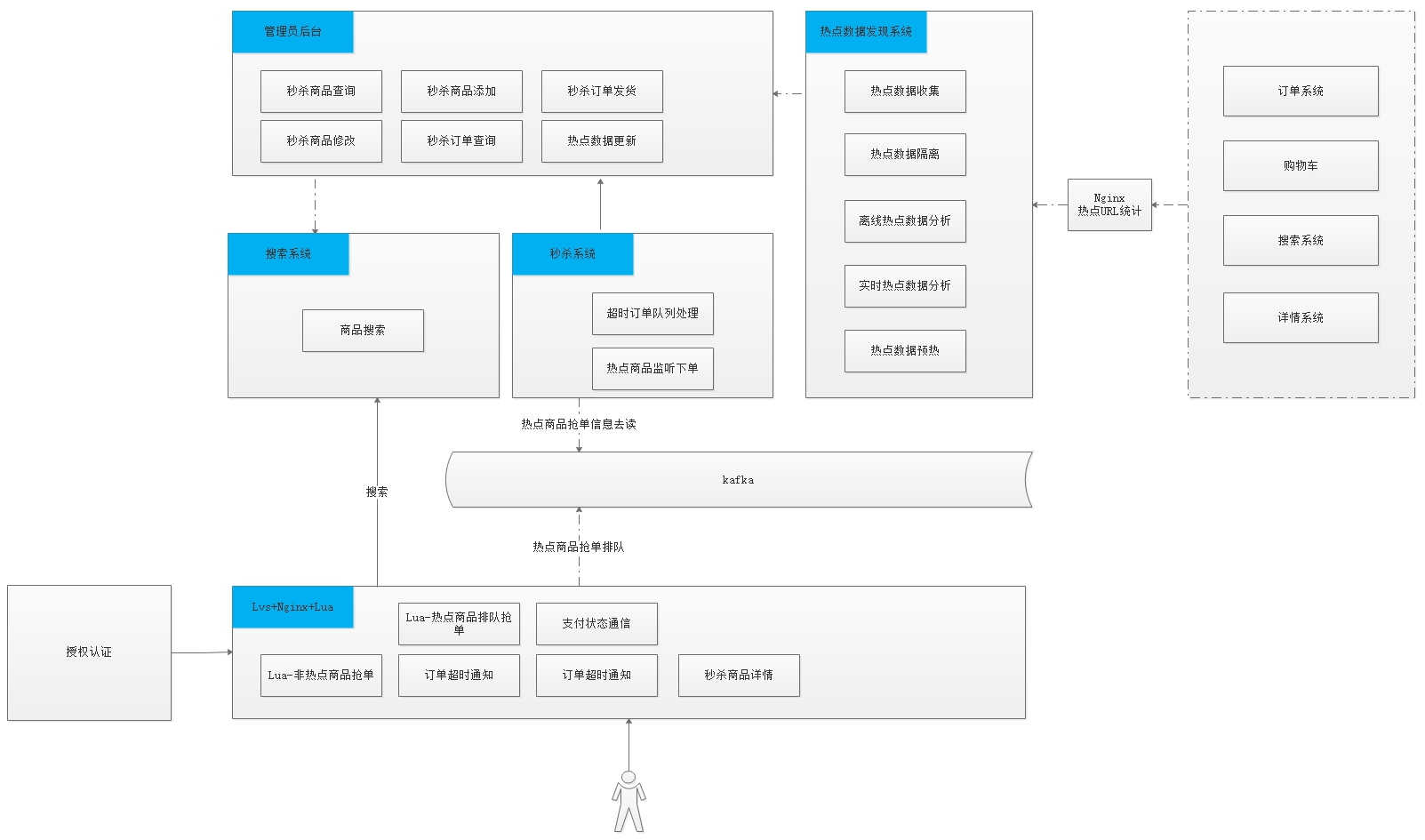
# 业务概述

## 需求概述

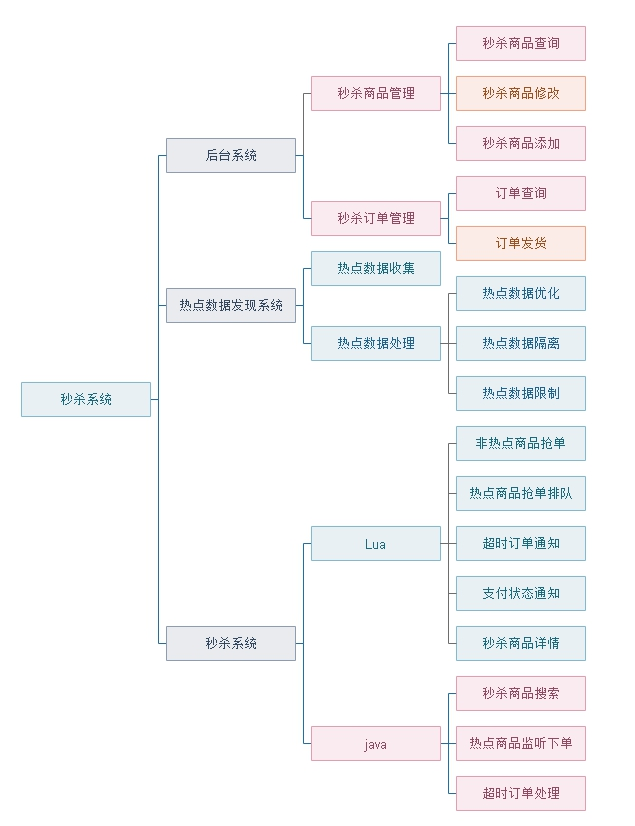
电商项目中，秒杀属于技术挑战最大的业务。后台可以发布秒杀商品后或者将现有商品列入秒杀商品，热点分析系统会对商品进行分析，对热点商品做特殊处理。商城会员可以在秒杀活动开始的时间内进行抢购，抢购后可以在线进行支付，支付完成的订单由平台工作人员发货，超时未支付订单会自动取消。

## 系统功能

### 核心业务流程



### 功能模块划分



注：以上功能点参考**需求**后持续修订

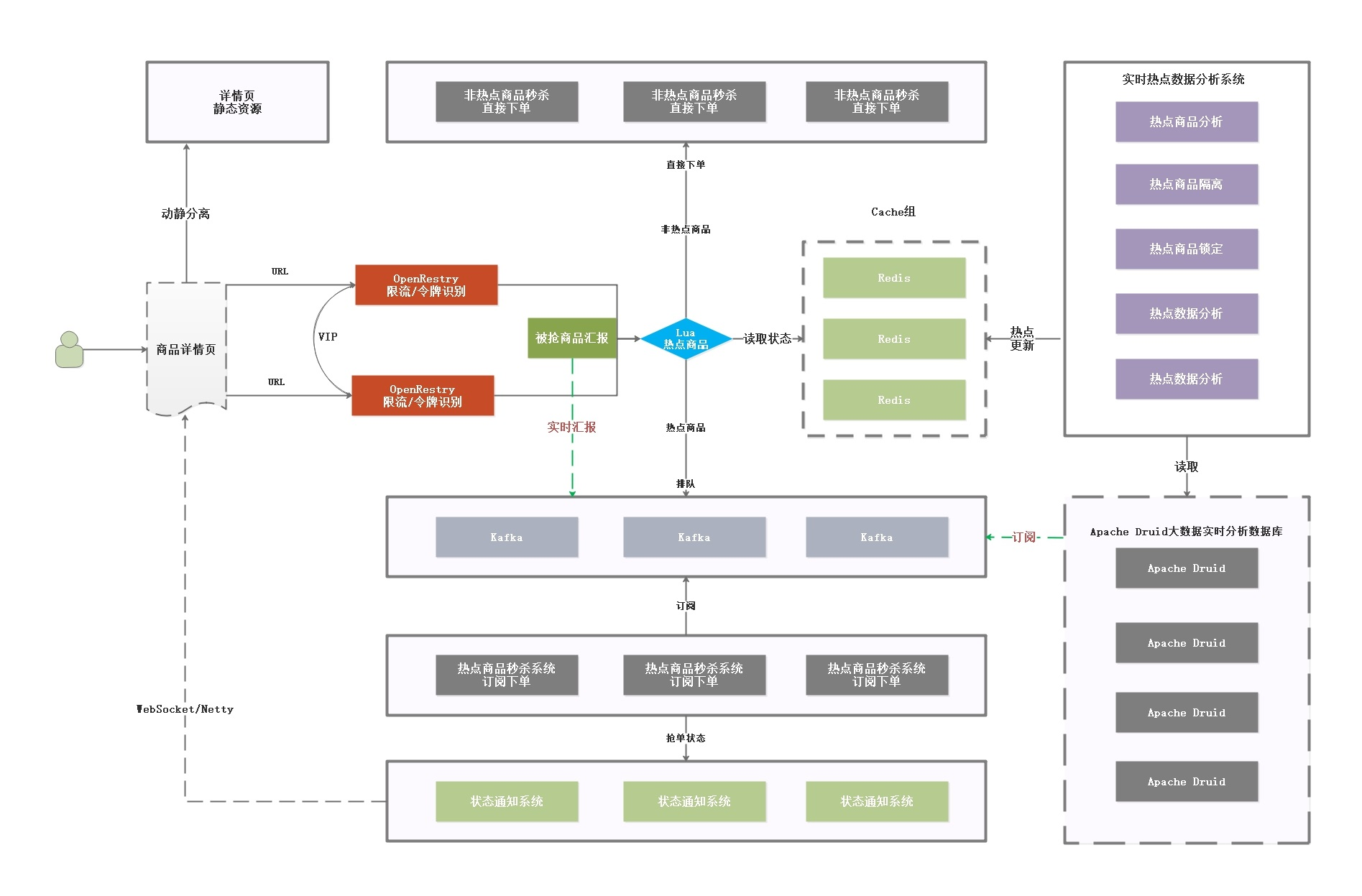
## 设计约束

在系统设计建设中，需遵守如下原则：

* + 开放性：基于业界开放式标准，为未来的业务发展奠定基础；
  + 扩展性：方便扩展设备容量和提升设备性能；具备支持业务处理的灵活配置，业务功能的重组与更新的灵活性
  + 安全性：系统访问及系统存储应安全。
  + 一致性：按照公司要求使用的技术、命名风格进行开发完成。

# 系统结构

## 技术架构图



### 项目目录结构说明

|  |  |  |
| --- | --- | --- |
| seckill-parent |  | 秒杀主工程 |
|  | config | 项目配置包 |
|  | controller | Rest Controller包 |
|  | Pojo | 数据库实体对象映射类包 |
|  | es | ElasticSearch对接封装包 |
|  | mapper | 数据库mapper包 |
|  | service | 业务逻辑处理接口及实现包 |
|  | util | 通用工具类包 |

# 技术体系

## 技术栈



## 技术亮点

### 基础

* 前后端分离架构，前端工程采用Vue.js+elementUI
* 使用Nacos作为配置中心来存储应用配置
* 基于SpringBoot开发实现
* ApacheDruid作为海量数据实时收集实时分析服务
* 将实时热点数据存入Redis提高查询效率
* 采用Mybatis对接Mysql数据库
* 采用SpringData ES来对接ElasticSearch的增加、更新、搜索(分页)
* 采用Lua实现冷热数据抢单程序隔离
* 采用Kafka实现抢单排队

### 实战

1. 全面讲解秒杀业务流程
2. 全面讲解秒杀系统架构
3. 对秒杀系统中超卖、高并发、热点数据发现、队列削峰、限流、集群、缓存安全策略、MQ数据安全、动静分离做全面讲解
4. 实时数据分析和离线数据分析讲解
5. 百万并发秒杀系统搭建+服务部署方案
6. 系统所使用的技术属于高阶技术

### 运维

* Mysql集群搭建
* Elasticsearch集群搭建
* Redis集群搭建
* Apache Druid集群搭建
* ElasticSearch集群搭建
* Docker使用
* Nacos集群搭建

# 业务逻辑设计

## 主要业务逻辑列表

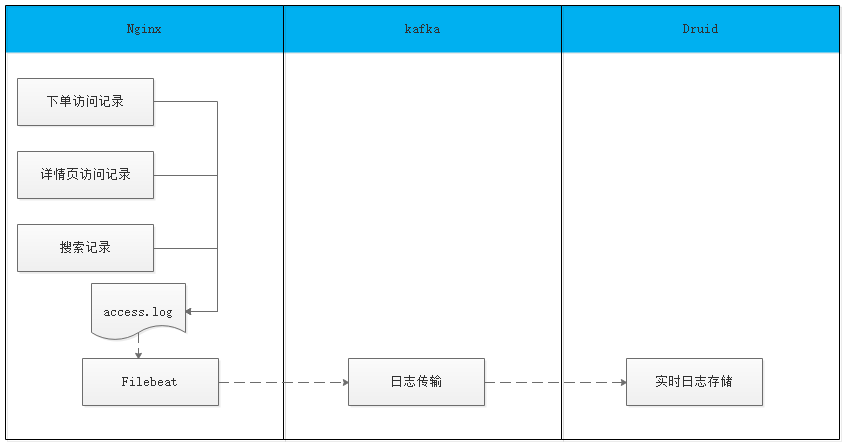
* 秒杀时间段管理
* 订单管理
* 用户管理
* 用户
* 首页
* 秒杀列表
* 秒杀详情页
* 立即购买
* 订单列表

## 主要业务逻辑设计

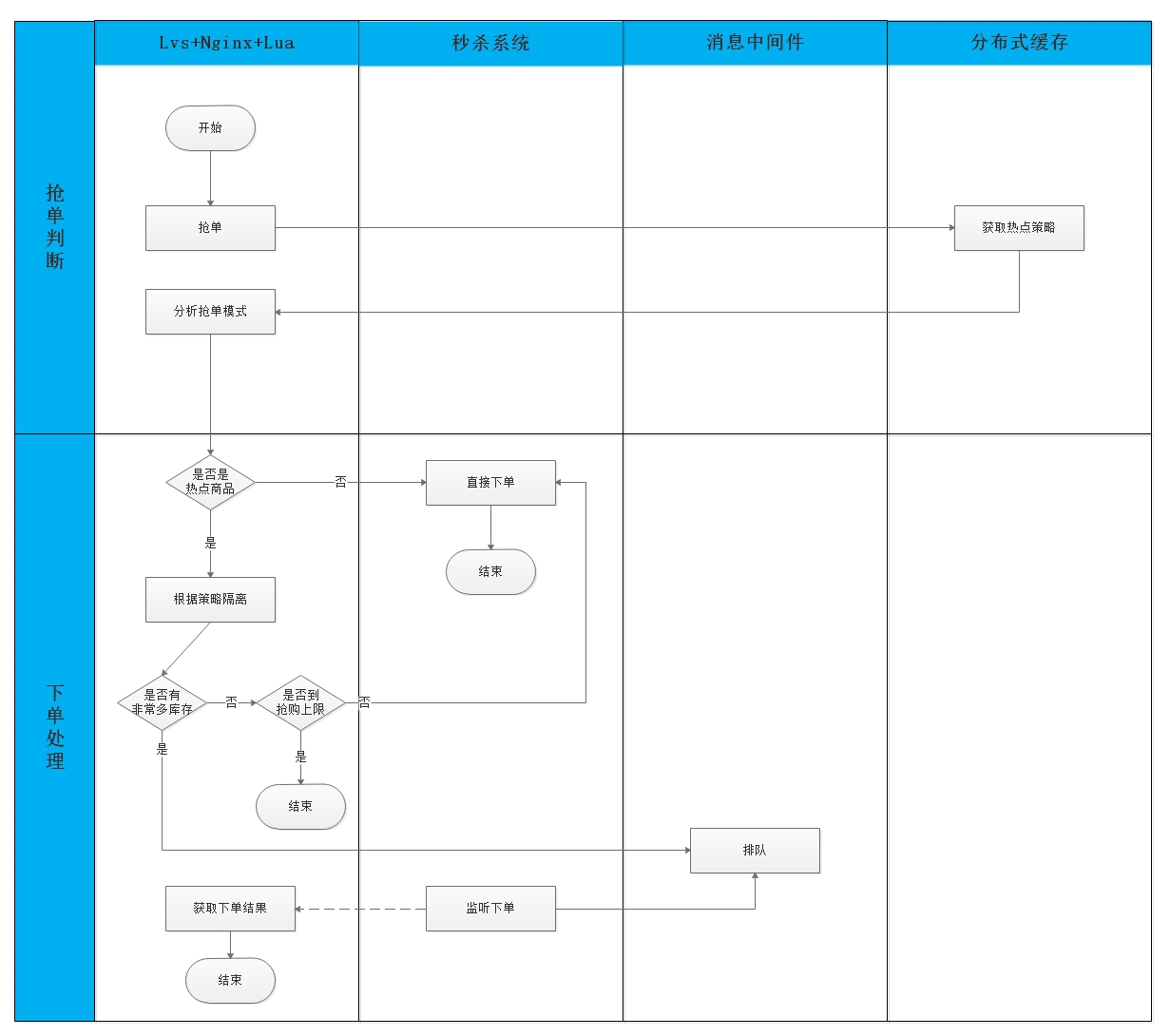
**注：目前此技术方案文档仅对部分重点业务逻辑进行设计、举例。**

### 用户登录（略）

### 实时热点数据收集



### 秒杀抢单



# 内部接口

## 接口列表

|  |  |  |
| --- | --- | --- |
| 秒杀后台 | 管理员 | 管理员登录 |
| 商品管理 | 商品列表 商品多条件搜索 |
| 活动管理 | 活动列表 活动多条件搜索 添加活动 编辑活动 删除活动 商品设置：  商品列表  添加商品  编辑秒杀信息  删除商品 |
| 秒杀时间段管理 | 秒杀时间段列表 秒杀时间段条件搜索 添加时间段 时间段编辑 时间段删除 时间段启用 |
| 订单管理 | 订单列表 订单列表条件查询 |
| 用户管理 | 用户列表 用户条件欧苏苏 |
| 秒杀H5前台 | 用户 | 用户登录 用户退出 |
| 首页 | 当前秒杀头版展示 |
| 秒杀列表 | 不同时段秒杀商品列表 秒杀倒计时功能 |
| 秒杀详情页 | 详情展示 倒计时功能 |
| 立即购买 | 下单功能 订单详情填写页 支付成功页 |
| 订单列表 | 全部订单 待付款订单 待收货订单 已完成订单 未支付订单支付页 |

# 数据库设计

## 实体关系图(E-R)

