**订单中心（支付） - 系统概要分析设计**

|  |  |  |  |
| --- | --- | --- | --- |
| 文档版本号： | *V1.0.0* | 系统名： | *订单中心* |
| 文档密级： | *保密* | 归属部门/项目： | *平台技术部* |
| 编写人： | *赖金满（Nick Lai）* | 编写日期： | *2017-10-14* |

**万普拉斯（深圳）有限公司 版权所有**

**内部资料 注意保密**

修订记录：

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订人** | **修订日期** | **修订描述** |
| V1.0.0 | 赖金满 | 2017-10-14 | 创建文档 |
|  |  |  |  |
|  |  |  |  |

**＊动作类型：批准、审核、通知、归档、参与会议，其它（请说明）**

**目录**

[1. 摘要 5](#_Toc478544583)

[编写目的 5](#_Toc478544584)

[项目背景 5](#_Toc478544585)

[任务概述 5](#_Toc478544586)

[目标人员 5](#_Toc478544587)

[规范与约定 6](#_Toc478544588)

[术语与缩略语 6](#_Toc478544589)

[参考资料 6](#_Toc478544590)

[2. 系统分析设计 6](#_Toc478544591)

[总体架构分析 6](#_Toc478544592)

[用例图 7](#_Toc478544593)

[业务用例图 7](#_Toc478544594)

[系统用例边界 7](#_Toc478544595)

[部署架构 7](#_Toc478544596)

[领域模型 8](#_Toc478544597)

[状态机 8](#_Toc478544598)

[核心业务规则 8](#_Toc478544599)

[业务规则1 8](#_Toc478544600)

[现有系统影响分析 9](#_Toc478544601)

[系统1 9](#_Toc478544602)

[3. 接口设计 9](#_Toc478544603)

[系统1 9](#_Toc478544604)

[接口1 9](#_Toc478544605)

[4. 数据库设计 10](#_Toc478544606)

[表1 10](#_Toc478544607)

[5. 非功能性特性设计 11](#_Toc478544608)

[可靠性 11](#_Toc478544609)

[可扩展性 11](#_Toc478544610)

[可运维 12](#_Toc478544611)

[安全性 12](#_Toc478544612)

[可测试性 12](#_Toc478544613)

[可维护性 12](#_Toc478544614)

[6. 其他 13](#_Toc478544615)

[发布回滚 13](#_Toc478544616)

[数据迁移 13](#_Toc478544617)

# 摘要

## 编写目的

此文档主要提供订单中心支付相关功能功能技术方案，主要提供：

* 订单中心与支付系统的系统结构技术方案。
* 订单中心与支付系统的支付流程技术方案。
* 订单中心与支付系统的订单风控交互流程技术方案。
* 订单中心与支付系统人工审核流程技术方案。

主要为后期的订单中心方案设计、产品开发和测试人员提供指导和帮助。

## 项目背景

因为没有，所以要有！

## 任务概述

* 商城订单通过订单中心和支付系统完成支付。
* 商城已支付订单通过订单中心和支付系统完成部分或全额退款。
* 商城和订单中心支付异常处理和人工干预。
* 通过订单中心人工干预风控。
* 通过订单中心处理重复退款。
* 通过订单中心查询订单、交易和相关风控数据。
* 常见支付异常和人为操作指引。

## 目标人员

项目经理、架构师、产品经理、系统分析设计师、开发人员、测试人员和客服。

## 规范与约定

* 订单中心向其它模块提供统一 RPC 接口和对接规范。
* 订单中心提供 RPC 接口必须满足当前业务需求，同时可以水平扩容。

## 术语与缩略语

*描述与本文档相关的业务或技术上的术语，如：*

|  |  |  |
| --- | --- | --- |
| **缩略语/术语** | **全称** | **说明** |
| OC | Order Center - 订单中心 | 保存和提供订单数据业务系统。 |
| Pay | Pay – 内部支付系统 | 提供与第三方完成交易接口、风险控制、和交易信息获取等功能的系统。 |
| PP | Payment Platform - 第三方支付服务平台 | 提供与第三方支付平台、风控平台对接服务。 |
| RCP | Risk Control Platform – 第三方风险控制平台。 | 第三方提供的对交易进行风险控制服务平台。 |

## 参考资料

* 单个IPN对应不同的订单状态：http://cf.onepluscorp.cn/pages/viewpage.action?pageId=18550605

# 系统分析设计

## 系统设计目标

订单中心对支付业务需求设计需要同时满足海外和国内支付、风控、退款和人工审核需求：

通过接口获取订单信息：? TPS

订单中心处理IPN：? TPS

## 总体架构分析

* 用户和客服下单不在保存在 Store，数据仅保存一份在订单中心，Store、Pay、GSP等系统获取订单只能通过订单中心，避免数据一致性复杂性和冗余性。
* 支付系统对订单的交易变动，由原来的通知Store，改为通知订单中心，Store通过订单中心获取到订单交易的当前状态，避免链路过长带来的不稳定性。
* 客服在GSP系统取消订单，由原来的通过Store取消订单，改为直接通过订单中心进行取消订单，理由同上。
* 客服在GSP发起订单退款（不取消订单），由原来的通过Store发起退款申请，改为直接通过订单中心退款，由订单中心统一控制订单交易的可操作性和订单管控。
* 客服通过GSP进行异常交易的人工审核处理，理由同上。

## 用例图

### 业务用例图

* 用户下单



* 客服代下单



* 用户支付订单



* 用户取消订单



* 客服退款



* 人工审核异常交易



### 系统用例边界

*系统用例主要描述全局的业务用例在各个系统间的分布，以明确业务在系统间的流转，以及确认系统边界。*

## 部署架构



## 领域模型

*不同于数据库表结构，领域模型适用于业务层，是对数据库数据在业务上的加工，比如商品描述和商品基本信息是属于一个领域模型，但在数据库中是两个表，因为商品描述通常是text类型，数据量比较大，和主表在一张表会拖慢主表的查询。请使用uml图中的类图来描述领域模型。*

### 状态机



## 核心业务规则

### 支付&审核规则



支付成功的判断唯一标准：至少有一笔成功的交易，且成功的交易金额至少要大于或等于订单总金额。

是否能支付订单，先由订单中心进行判断，然后由支付系统判断（是否已经支付）。

### 退款流程



订单中心在操作退款前，除了要判断是否具备退款条件，还要先取消发货，才能进行退款，否则可能存在钱退了，货还继续发的风险。

## 现有系统影响分析

列出本项目涉及的对现有哪些系统、哪些业务的影响分析。需要涉及配合改造或者部署有变更的才是有影响，比如jar包升级不兼容，需要外围系统配合改造，或者某个服务实现增加了判断，结果码有增加，需要外围系统配置文案，或者变更业务判断等。

### 系统1

# 接口设计

*描述系统对外发布的服务接口设计，包括内部系统间rpc服务、http服务，以及和第三方的服务对接，着重点在描述输入输出、调用方法，以及服务提供的性能指标。*

## 订单中心接口

### 订单数据获取接口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **服务(url)名称** | getOrder | | | |
| **服务描述** | 支付模块获取订单数据 | | | |
| **服务质量** | TPS 2k  Time < 50 ms | | | |
| **调用方法** | Dubbo RPC | | | |
| **输入** | **参数名称** | **必需** | **类型** | **说明** |
| OrderNo | 是 | Int | 订单号 |
| **输出** | **返回名称** | **必需** | **类型** | **说明** |
| *Result* | *是* | *com.oneplus.common.biz.Result* | *业务返回对象，成功将返回预扣库存的仓库、库区、配送方式以及运费* |

# 数据库设计

## 表1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名** |  | | | |
| **数据访问tps** | *读：100，写：10* | | | |
| **数据增长率** | *10w/天* | | | |
| **字段名** | **数据类型** | **主键** | **非空** | **字段描述** |
|  |  |  |  |  |
| **索引名** | **索引字段** | | | |
|  |  | | | |

# 非功能性特性设计

## 可靠性

*描述系统在可靠性上的分析和设计，系统在容错性上是如何处理的，故障恢复、服务降级的处理机制，以及在数据可靠性方面的考虑等。比如如何防止同步抢购订单时发生订单丢失的问题，需要考虑系统的自动处理故障以及自我恢复能力，以及订单数据存储的可靠性。*

## 可扩展性

*可扩展性涵盖了系统的编译期和运行期的设计，运行期如何具备水平扩展能力来提升系统性能，编译期如何设计良好的代码结构来适应业务在未来可预见时间内的变化。这里需要描述系统在可扩展性上的设计思路。*

## 可运维

*描述在提升系统运维方面的分析及设计，好的系统应该是尽可能自动化的完成业务，自身应该具备相当能力的容错处理，不需要运维人员介入处理；同时也尽可能是标准化的，方便运维人员部署，或者提供可视化的运维操作。*

## 安全性

*配合安全要求，阐述系统安全方面（例如XSS，SQL注入，DDOS，数据安全等）的设计。*

## 可测试性

*描述系统各个业务在可测试性上的分析，发布应该满足灰度要求，系统应该具备在生产环境的可测试性。必须等到某个时间点、修改操作系统时间，或者直接不可测，都不应该发生。*

## 可维护性

*可维护性包含可扩展性在编译期的设计，可维护性的系统必然具备可扩展性，此外，可读性、命名规范、包路径约定等也是可维护性的考量。*

# 其他

*描述其他关注点的分析，例如回滚方案，数据迁移方案等对系统设计的影响，系统设计如何处理才更好的支持。*

## 发布回滚

*这里需要描述的是系统如何设计才能更好的进行发布回滚，而不是描述回滚方案，设计合理可以确保回滚更易于处理。*

## 数据迁移

*和发布回滚类似，这里也是需要描述系统如何设计才能进行数据迁移，而不是说明迁移方案。*