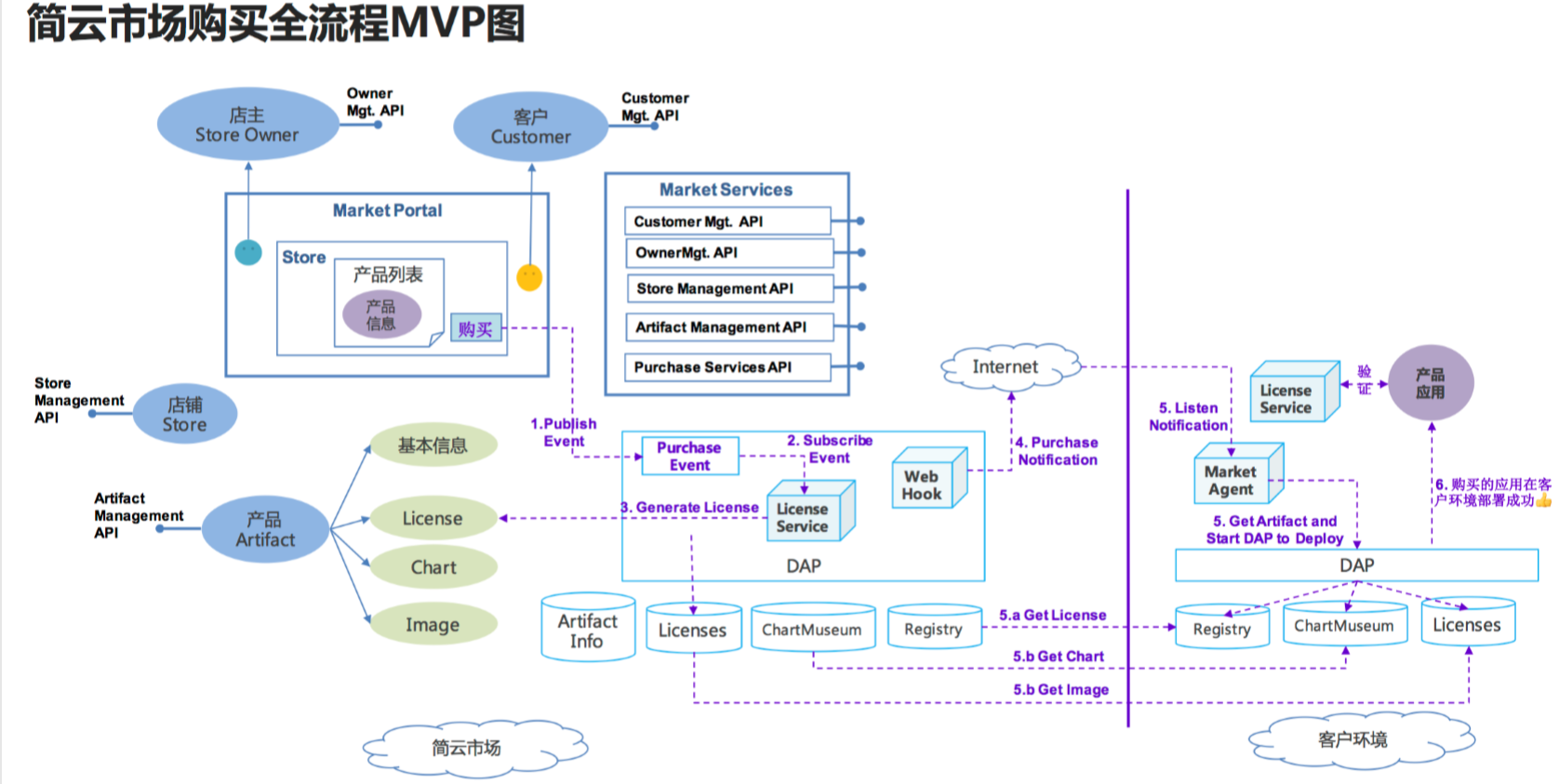
简云市场概要设计

#### 业务分析

简云市场是一套面向金融科技领域提供出各种组件和应用的开放市场。它彻底改变客户采购，购买，交付和使用软件服务和应用的模式和体验。通过应用license形式进行销售，能够将应用一键部署将到用户的私有云或公有云下，实现快速部署开箱即用，并给用户带来良好的使用体验 。

#### 业务架构



根据以上定义简云市场可分为如下三个部分：

1. 简云市场应用(market protal)

简云市场应用分为 简云市场以及市场管理两个应用。

简云市场是一个提供给互联网用户进行查看 购买简云应用产品的门户应用 。

简云市场管理是针对门户内容 以及订单 业务管理的应用 。

1. 简云市场服务(market service)

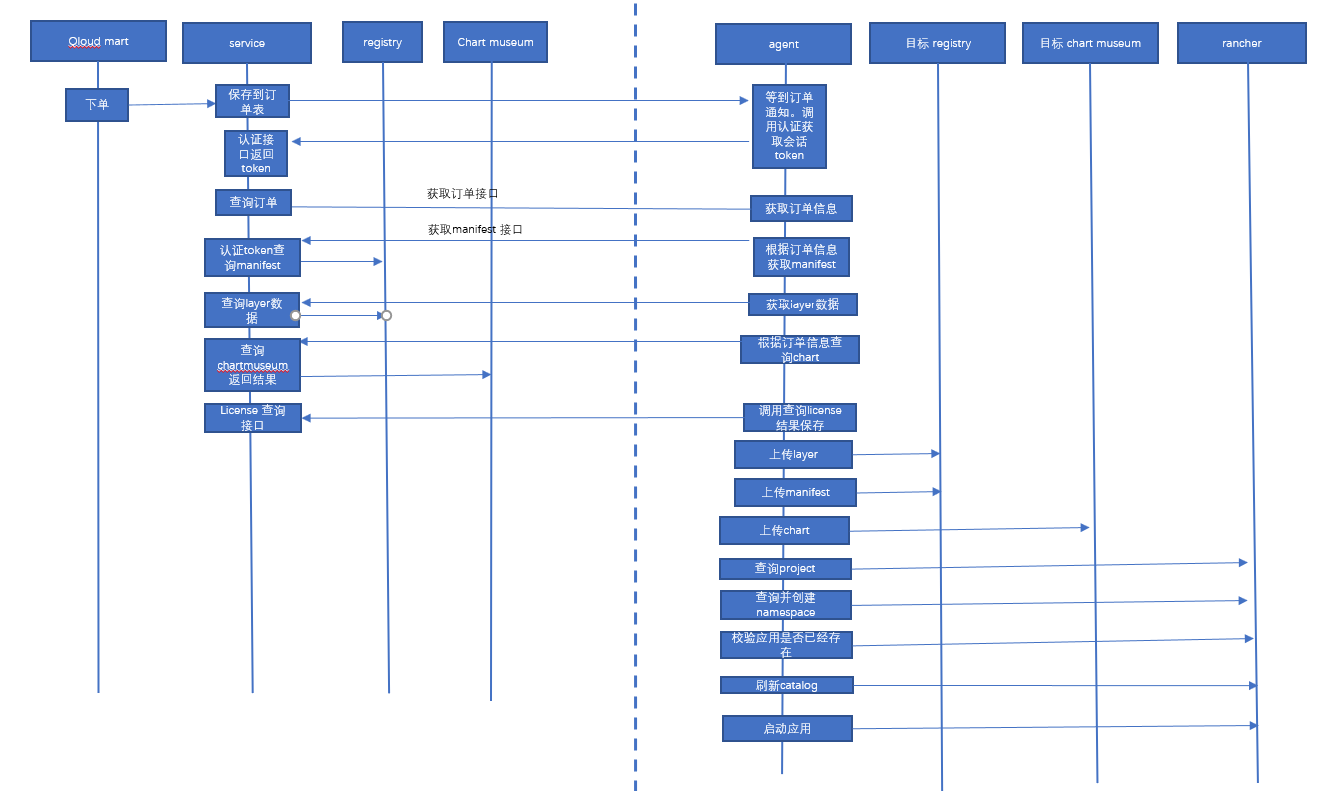
简云市场服务是提供一套简云市场所需要的API的微服务平台。它主要提供如下服务:

1. 订单服务
2. License同步
3. Image同步
4. Chart 同步
5. Agent 认证
6. 简云市场应用代理服务(market agent service)

简云市场应用代理服务，部署在用户环境下。通过轮询获取购买通知。当得到购买通知后远端拉取简云市场对应商品的image镜像。以及chart 文件。调用rancher接口实现自动部署。

#### 三、 业务流程

##### 流程说明



1. 客户可以浏览 简云市场中已经发布应用，决定购买其中某一个商品。当用户点击购买，并支付成功后。市场服务后台 (market service)得到购买event 事件后 对购买信息进行确认根据订单商品属性生成license。
2. 保存生成的license 和订单信息到数据库中。
3. 市场服务后台 (market service) 当有订单时发送订单通知，通知 agent拿去订单 。
4. Agent得到通知后 先请求market service 认证agent并获取会话token。
5. agent 拿到token 请求查询订单接口，获取到要发布的订单 (客户ID、商品ID、商品名称、商品license、chart路径、image路径)
6. Market agent 获取到以上信息后，根据订单号创建本地订单文件。
7. 请求license 接口获取license并将license保存在License文件中。
8. Market agent 请求market service chart接口和image接口，将chart和 image 文件拉取到用户环境。
9. Market agent 获取到 charts 和images 调用本地registry 和chartmuseum接口 上传image 和chart 文件。
10. Market agent 根据订单物流信息中项目名称查询Rancher中是否存在， 如果不存在则报错并记录到订单文件中。
11. Market agent 查询namspace 是否存在，如果不存在则创建namespace。
12. 查询project ，namespace 下是否已经运行应用，如果已经运行则报错并更新订单文件。
13. 刷新rancher catalog
14. 调用rancher启动应用接口，启动应用，如果报错，更新到订单文件。

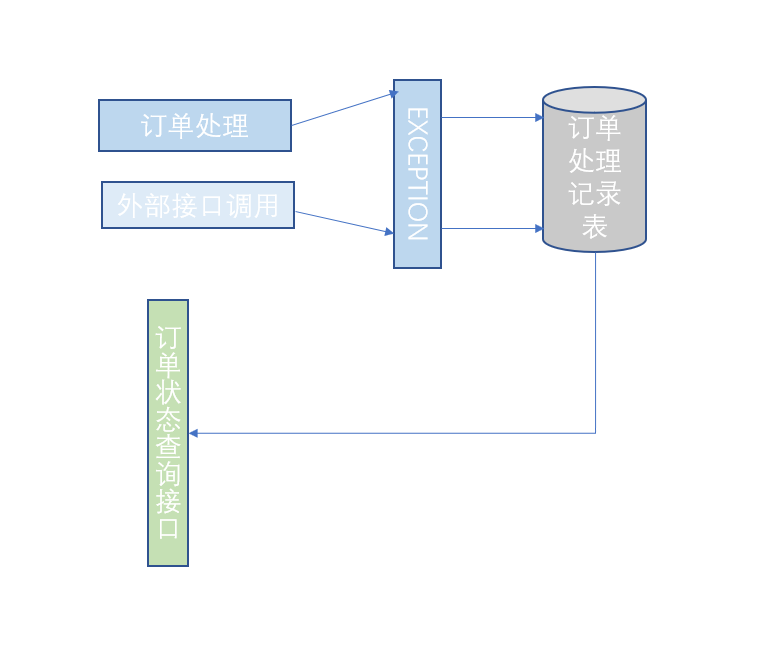
##### Market agent 调用其它应用的接口

|  |  |  |  |
| --- | --- | --- | --- |
| **应用** | **接口** | **描述** | **状态** |
| market Service | agent 认证接口 | 当得到通知后确认是自己环境下的应用请求market service 获取认证token | 未开发 |
| 订单查询接口 | 获取需要部署的订单信息 | 需改造 |
| license 查询接口 | 获取订单下产品的license 集合 | 已开发 |
| image manifest 查询接口 | 获取订单下产品 image的manifest 如果是多个产品或多个image 多次调用 | 未开发 |
| layer 下载接口 | 根据manifet获取layer 数据 | 未开发 |
| chart 文件查询接口 | 根据订单信息获取chart文件 | 未开发 |
| registry | image layer上传接口 | agent 上传下载的layer 数据 | 已开发 |
| manifest 上传接口 | agent 上传image的 manifest | 已开发 |
| chart museum | chart上传接口 | 上传chart文件 | 已开发 |
| rancher | 查询projects | 查询订单信息中的project 属性的值是否再目标环境下存在 | 已开发 |
| 查询namespace | 查询订单信息中的namespace 属性的值是否再目标环境下存在 | 已开发 |
| 新建namespace | 如果namespace 不存在则新建 | 已开发 |
| 查询应用已经部署接口 | 如果应用已经村则则不部署 | 已开发 |
| 刷新rancher catalog 接口 | 刷新rancher 商店catalog | 已开发 |
| 启动应用接口 | 启动已经上传chart的应用 | 已开发 |

##### Market service 调用其它应用的接口

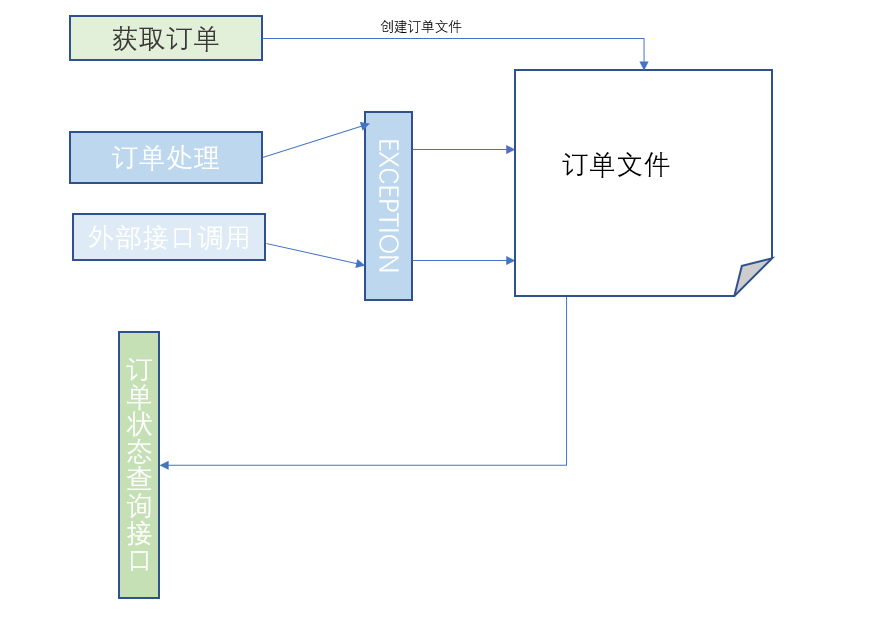
|  |  |  |  |
| --- | --- | --- | --- |
| **应用** | **接口** | **描述** | **状态** |
| market agent | 订单通知接口 | 通知agent 去获取订单 | 未开发 |
| registry | image查询接口 | 查询image 是否存在同时获取tag | 未开发 |
| tag 校验接口 | 查询产品image 的tag是否存在 | 未开发 |
| manifest 查询接口 | 获取manifest 信息 | 未开发 |
| layer下载接口 | 根据agent 发送的degist获取layer 数据 | 未开发 |
| chartMuseum | chart 下载接口 | 根据agent 发送的chart 文件名 获取chart数据 | 未开发 |

##### Market service 异常处理



1. 当用户下单的订单再market service 处理出现异常时 我们会将异常信息（错误代码，错误信息）保存到订单处理日志表中，并同时修改订单状态。
2. 提供订单处理状态查询接口，提供用户查询订单的状态以及处理信息

##### Market agent异常处理



1. Agent 获取到订单后创建订单文件。
2. 订单处理时 将每一步操作记录再订单文件中
3. 在出现异常的时候也将异常信息 记录在订单文件中
4. 提供订单状态以及结果的查询接口

#### 四、接口设计

##### Market service 接口

|  |  |  |
| --- | --- | --- |
| **market service API** | | |
| **接口名** | **描述** | **状态** |
| 下单接口 | 前端下单到market service 进行订单处理 | 已有 |
| 订单信息查询接口 | agent 请求market service 获取订单信息，请求信息中有agent的相关认证信息 | 需开发 |
| image manifest 查询接口 | agent 根据订单信息获取获取image的name以及tag 再请求该接口获取image 的 mainfest 信息 | 需开发 |
| layer 下载接口 | agent 根据manifest 信息获取layer 的degist 根据degist 下载layer 数据 | 需开发 |
| chart 文件查询接口 | agent 根据订单中chart 文件名 请求market service 获取chart 文件 | 需开发 |
| license 查询接口 | agent 根据订单信息 获取订单中每一个商品的license 文件 | 需开发 |
| agent 认证接口 | agent 发送agent 相关信息从market service 获取到授权的会话token | 需改造 |
| 订单状态查询接口 | 前端或者第三方可以查询market service 所处理的订单状态 | 需开发 |
| agent新增接口 | 前端通过界面 发送agent 配置属性并再后台新增一个agent节点 | 需开发 |
| agent 查询接口 | 通过前端界面查询agent 相关的列表信息 | 需开发 |
| agent 修改接口 | 通过前端界面删除或修改agent | 需开发 |

##### Market Agent接口

|  |  |  |
| --- | --- | --- |
| **接口名** | **描述** | 状态 |
| 订单通知接口 | market service 调用该接口通知agent 去获取订单 | 需开发 |
| 产品license查询接口 | 应用程序通过调用该接口获取license的相关信息确保应用是否有效 | 已有 |
| 部署状态查询 | market service 可以根据该接口查询订单中相关应用的部署状态。 | 需开发 |