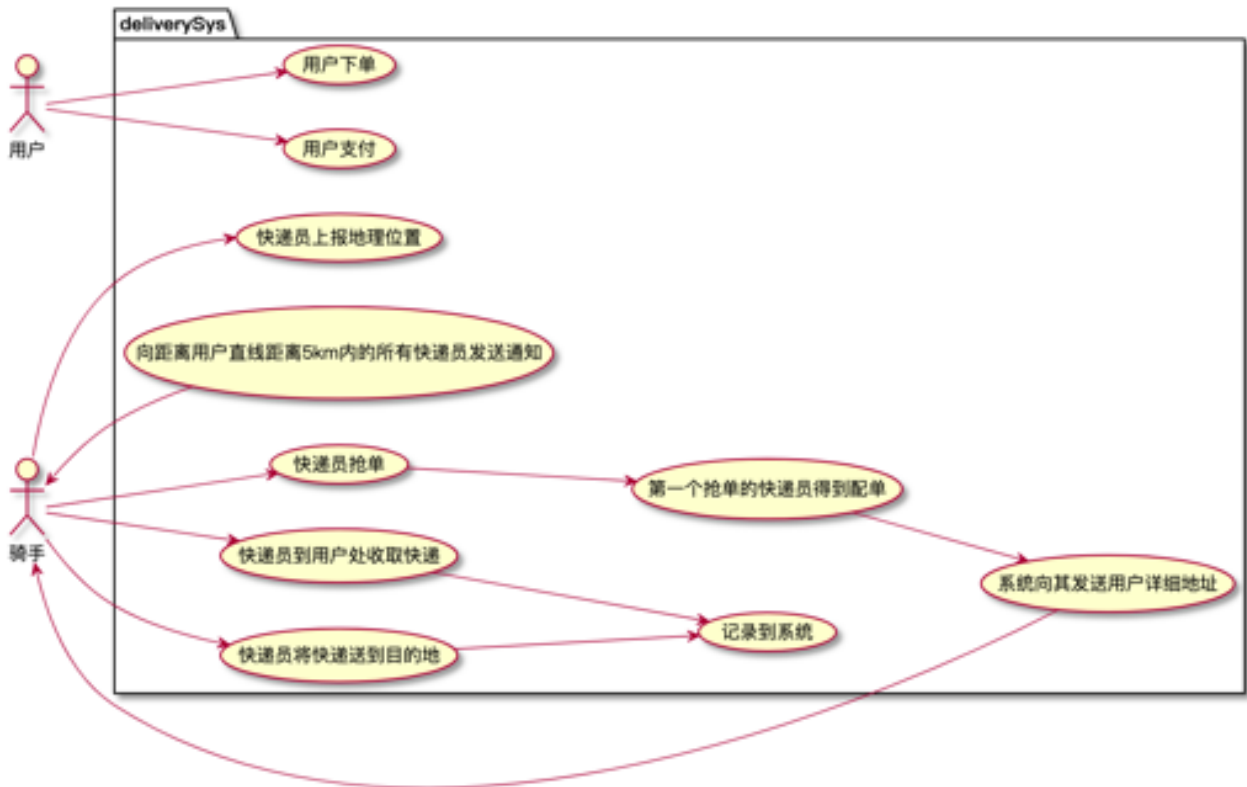


通达架构

1 设计概述

通达是某上市公司全资投资成立的一家物流快递公司，主要进行同城快递业务。

1.1 功能概述



系统包括的功能如下：

- 用户通过app发起快递下单请求并支付
- 快递员通过自己的App上报自己的地理位置，每30秒上报一次
- 系统收到快递请求后，向距离用户直线距离5km内的所有快递员发送通知
- 快递员需要进行抢单，第一个抢单的快递员得到配单，系统向其发送用户详细地址
- 快递员到用户处收取快递，并记录到系统中：已收件
- 快递员将快递送到目的地，并记录到系统中：已送达

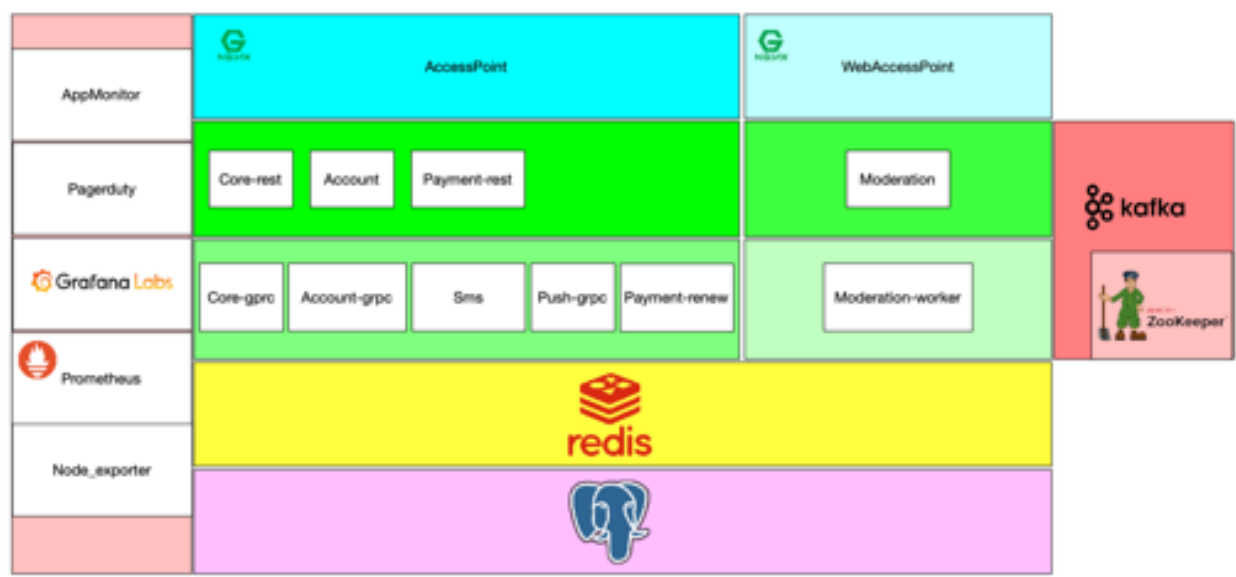
1.2 非功能约束

预计上线后三个月日单超过1万，一年日单超过50万。

- 配对快递员性能目标：平均响应时间<100ms，95%响应时间<150ms，单机QPS > 1000；
- 下单性能目标：平均响应时间<100ms，95%响应时间<150ms，单机QPS > 1000；
- 系统核心功能稳定性目标：>99.99%
- 系统安全性目标：系统可拦截各种非法请求，恶意攻击，密码数据散列加密，客户端数据HTTPS加密，外部系统间通信对称加密；
- 数据持久化目标：>99.99999%

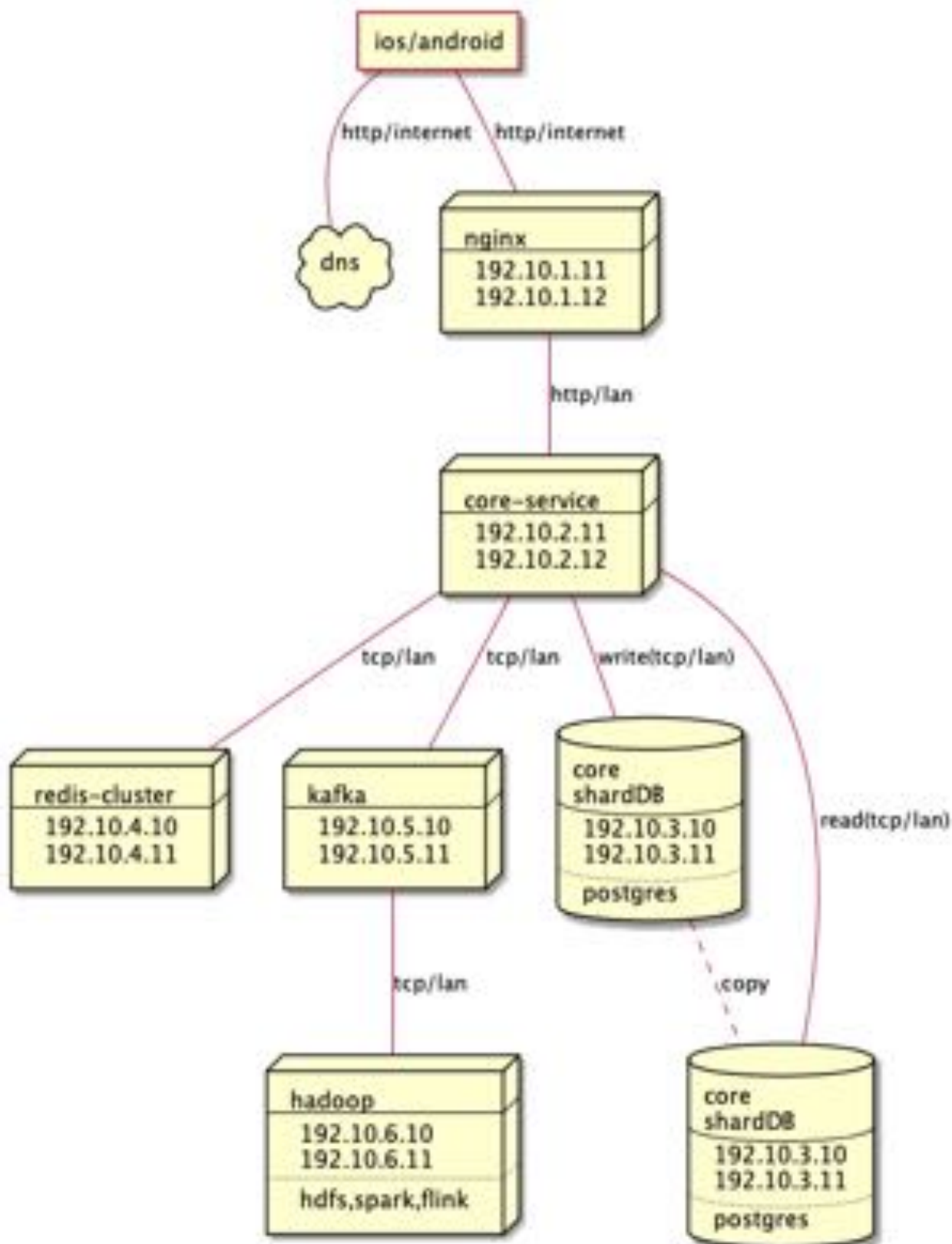
2 系统部署图与整体设计

2.1 系统架构图



图中左侧淡红色部分是监控、报警模块，使用Node_exporter采集各个机器的基础监控数据（包括CPU、内存、磁盘、网络）。Prometheus负责存储采集的监控数据。Grafana负责展示自定义的监控数据报表。PagerDuty基于Prometheus采集的监控数据配置报警规则，触发电话、短信等报警。图中顶部是接入层，使用Nginx做为gateway，负责鉴权、限流、负载均衡等策略。接入层下面是内部服务，左边是面向用户端的服务，右边是面向内部用户的管理平台；第一层是Rest服务，负责返回前端展示需要的数据；第二层是一些内部的grpc、worker等内部服务，负责为Rest服务提供数据，以及消费事件。再下层是Cache，负责加快用户访问。最下层是DB，用于持久化用户数据。最右侧是Kafka集群，负责接收和分发事件。

2.2 系统部署图

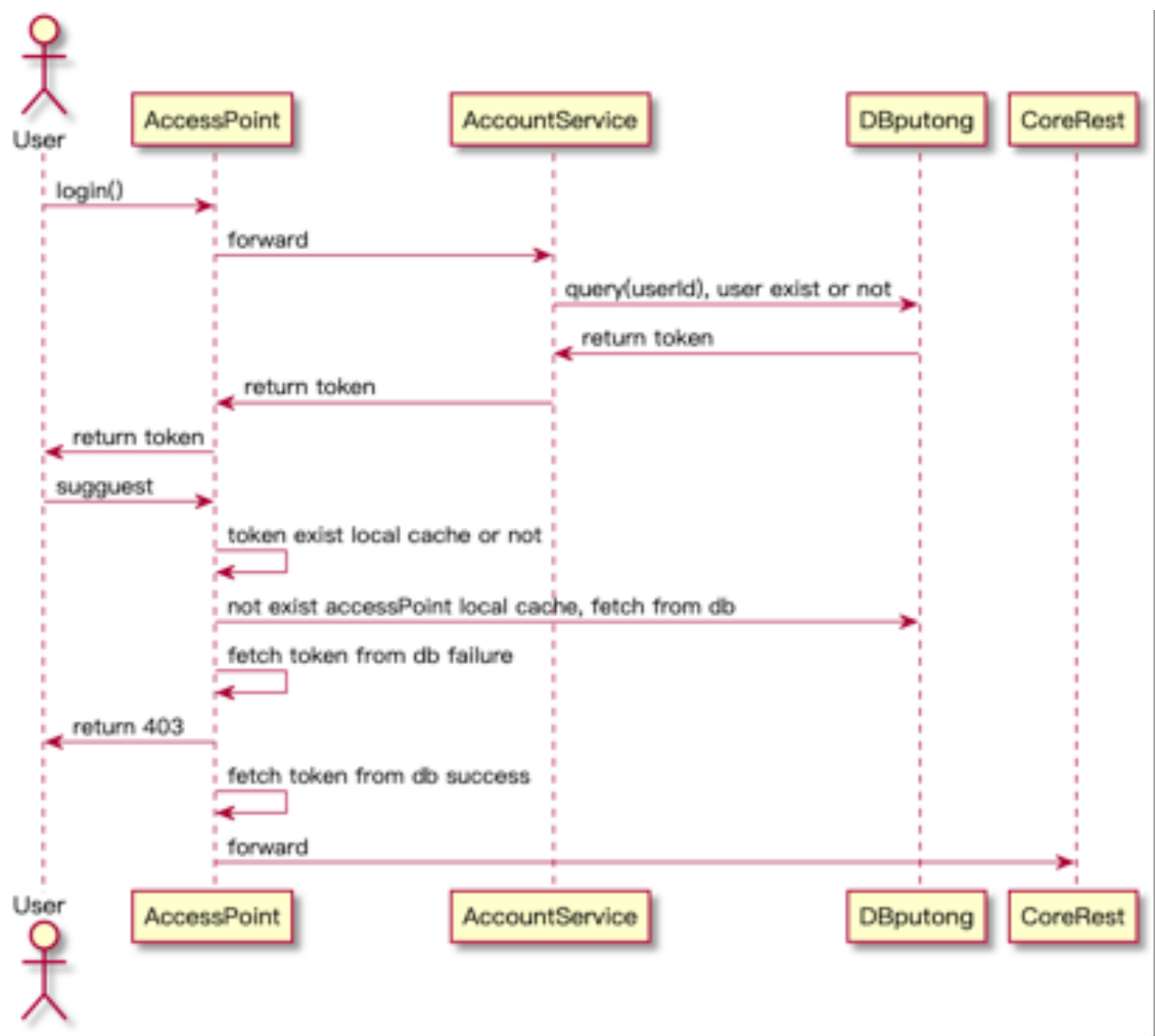


上图展示了系统的整体的部署图，ios/android通过公网访问dns，拿到后端接入层的ip，前端发送的请求首先到达gateway，此系统使用nginx做gateway，负责鉴权以及负载均衡，nginx将流量转发到核心服务core-service负责逻辑处理，使用redis-cluster做为分布式缓存使用，使用kafka做为消息队列，负责内部服务通信，以及和大数据hadoop交互。使用postgres存储业务数据，数据库一主一从，主库负责写请求，从库负责读请求。

3 接入层子系统设计

接入层使用Nginx做为gateway，负责鉴权、限流、负载均衡等策略。

3.1 鉴权时序图



4 Core子系统设计

4.1 下单抢单场景的业务活动图

