深入咚咚服务端架构

胡 峰

目录

- 架构演化
- 架构分解
- 实践总结

1. 架构演化

罗马不是一天建成的

单一应用：复杂的怪兽系统

（微）服务化

- 按业务能力来划分服务与组织团队：任何设计系统的组织，最终产生的设计等同于组织之内、之间的沟通结构。—— 康威定律
- 增加开发、测试、运维复杂度，但提升了并行性和可管理性

2. 架构分解

业务架构

一线客服

客服主管

京东用户

京东员工

客服业务

个人 / 员工业务

业务规则

接线

转接

邀评

挂起

运营监控

运营报表

日常管理

客服状态展示

咨询

留言

评价

私聊

群聊

桌面共享

语音通话

组织架构

排队规则

分配规则

分流规则

风控规则

升级规则

灰度规则

灰度规则

产品 / 研发业务

IM 基础业务

升级管理

用户反馈奖励

应用监控

数据管理

风控管理

配置管理

灰度管理

缓存管理

灾备切换

好友 / 联系人

群组

消息

临时会话

状态

咚咚产品

咚咚研发

内部业务系统

木桃 App

配送 App

应用架构

客服

员工

UI / 接入

WEB（首页、商品页、订单页、店铺页）

京东 APP SDK

PC 桌面

微信 / 手Q（m 版）

企业版 PC 桌面

企业版 WEB

企业版 APP

HAProxy

负载均衡

TCP （长连接）

HTTP （短连接）

路由服务

路由 Tracker

管控中心
(king)

业务服务

京东客服

商家客服（采销 / POP）

拍拍客服

咚咚管家

个人版

企业版

基础服务

认证服务

跟踪服务

资源访问服务

IM 基础业务服务

升级服务

风控服务

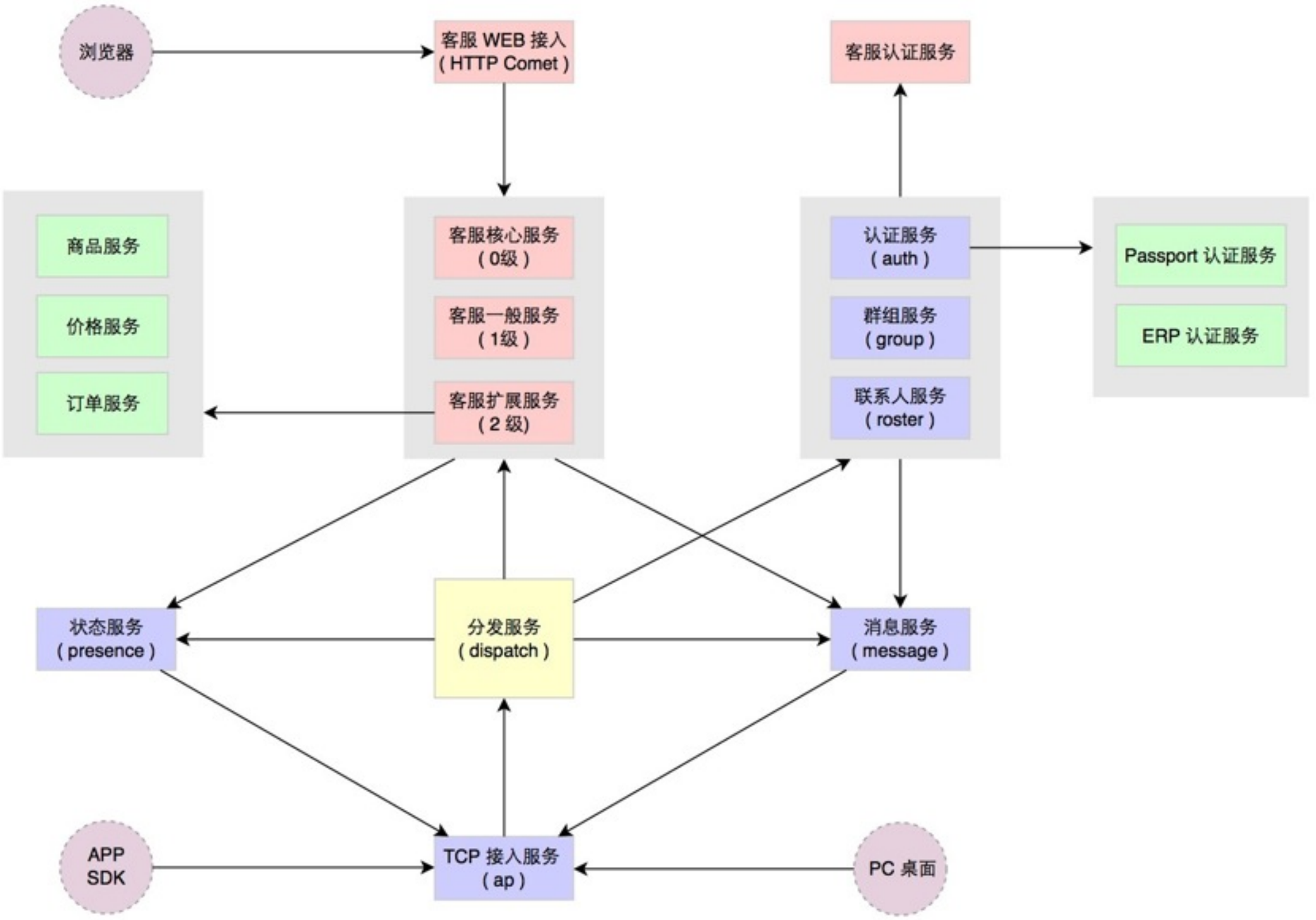
搜索服务

资源服务

MySQL

redis

MongoDB


浏览器
客服 WEB 接入
(HTTP Comet)
客服认证服务
商品服务
价格服务
订单服务
客服核心服务
(0级)
客服一般服务
(1级)
客服扩展服务
(2 级)
认证服务
(auth)
群组服务
(group)
联系人服务
(roster)
Passport 认证服务
ERP 认证服务
状态服务
(presence)
分发服务
(dispatch)
消息服务
(message)
APP
SDK
TCP 接入服务
(ap)
PC 桌面

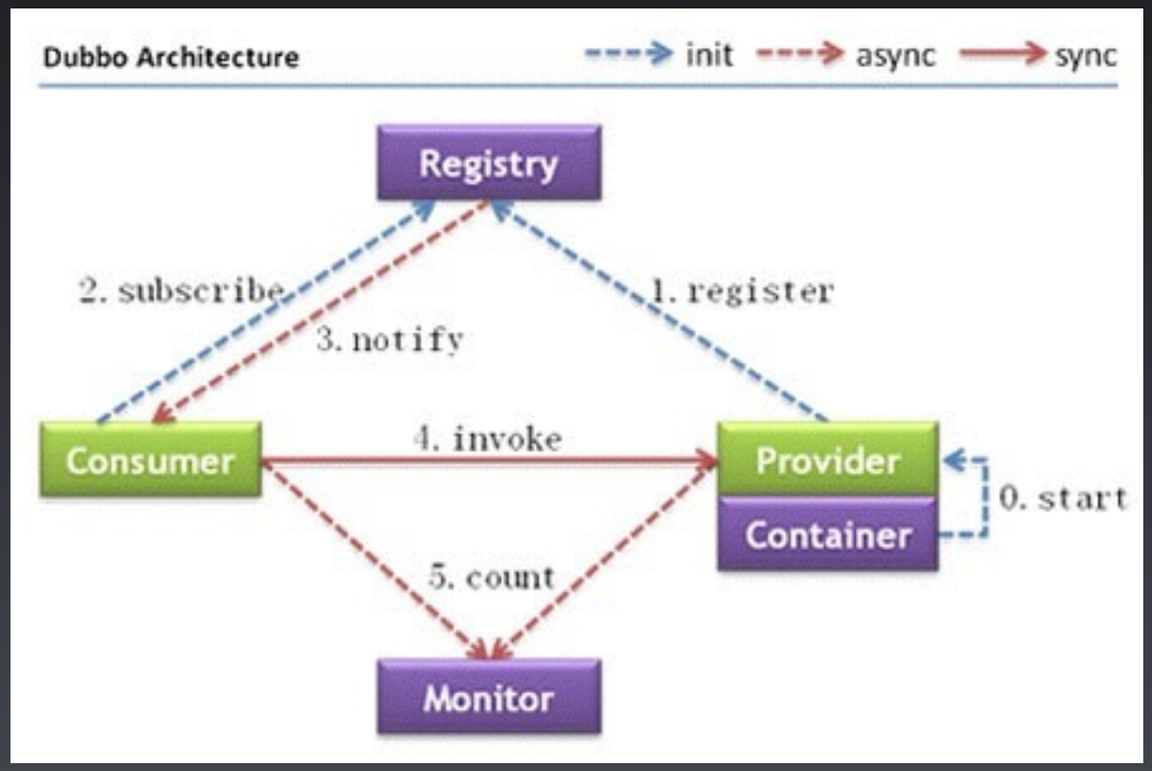
技术架构

redis

- 用作 Cache（在线状态）
- 用作 Buffer（聊天记录入库）
- 用作 DB（关系）

Dubbo

- 消息 RPC
- 使用简单
- 满足目前用户量级的性能需求



不能只开车，还得修路

Dec. 2013 ©unTed Photography

乘车人 - 业务需求方

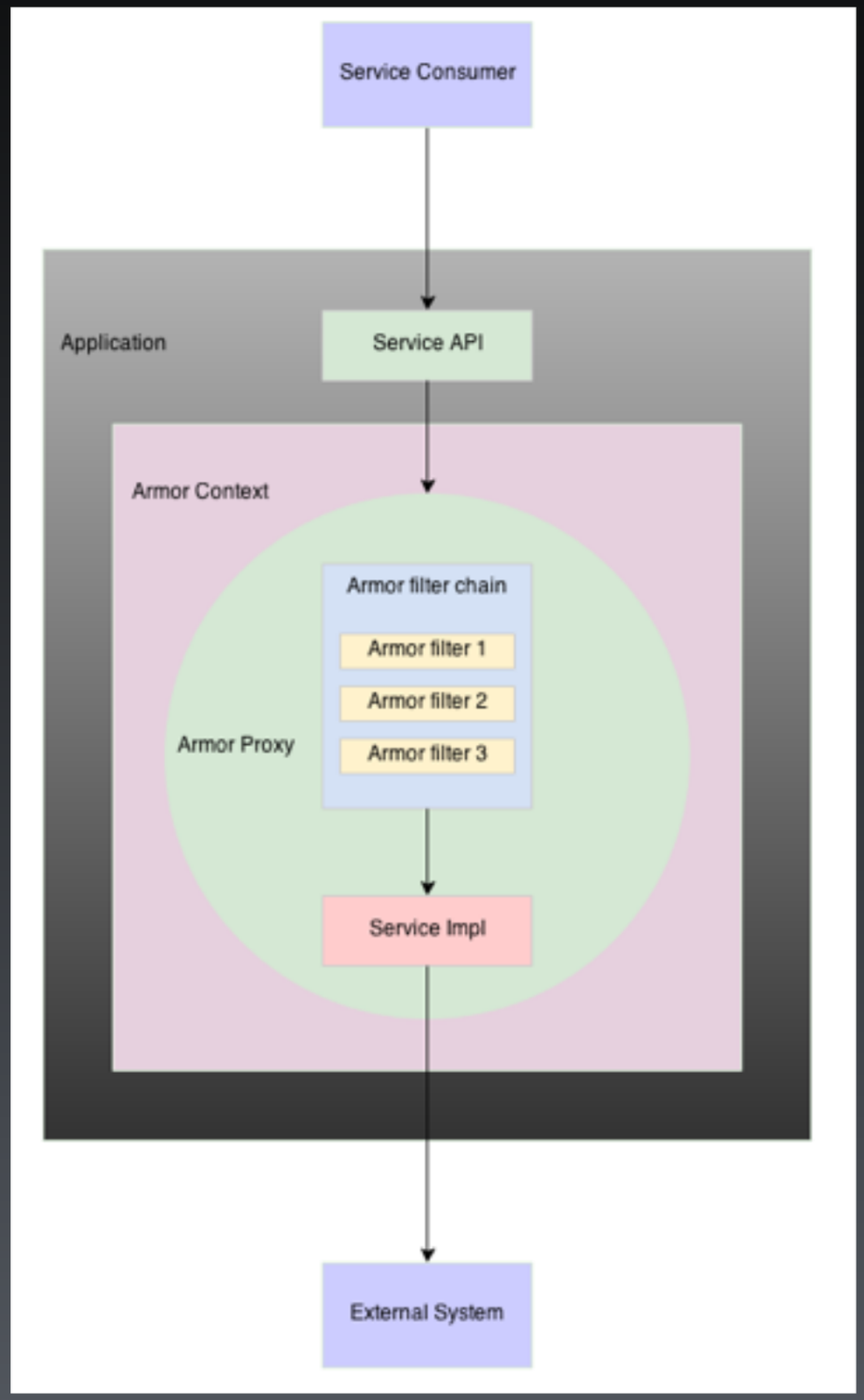
开车人 - 业务研发

修路人 - 基础研发

Craft

- 轻量 NIO, 支撑大量长连接
- Armor 服务增强

随业务发展而逐步完善的基础组件，更多请参见官网 http://craftcode.io



3. 实践总结

都是血的教训

原则

- Design for failure (容错设计)
- 发送时要保守，接收时要开放
- 协议和服务，版本向前兼容
- 轻前端，重后端，后端控制前端
- 隔离，进程和线程
- 保持透明

监控的维度

- 基础设施（IDC、网络）
- 基础平台（OS、JVM）
- 应用服务（线程池、对象池、RT、OPS）

开发、测试与运行

- 服务即产品（You build it，you run it）
- 自动化测试（功能、性能自动回归）
- 动态运行与运营

谢谢大家

@云下山巅

https://github.com/mindwind